

020-13638.1-A-E

12

Technical Report

**SIMULATION OF THE FLIGHT OF A SHORT
RANGE AIR DEFENSE SYSTEM ROCKET
BEFORE RADAR ACQUISITION**

by

John E. Cochran, Jr.
and
James R. Beaty

under contract with

U.S. Army Research Office
Research Triangle Park, North Carolina
Grant Number DAAG29-76-G-0069

**AUBURN UNIVERSITY
SCHOOL OF ENGINEERING**

ENGINEERING EXPERIMENT STATION

Mechanical Engineering Department

April 1977

Approved for Public Release
Distribution Unlimited

DDC
RECEIVED
JUL 27 1977
E

AD A 042036

AD WU.
DDC FILE COPY

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SIMULATION OF THE FLIGHT OF A SHORT RANGE AIR DEFENSE SYSTEM ROCKET BEFORE RADAR ACQUISITION.		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report. 1 October 1975-15 March 1977
7. AUTHOR(s) John E. Cochran, Jr. and James R. Beaty		8. CONTRACT OR GRANT NUMBER(s) DAAG29-76-G-0069 new
9. PERFORMING ORGANIZATION NAME AND ADDRESS Engineering Experiment Station Auburn University Auburn, Alabama 36830		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Research Office Post Office Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE April 1977
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) ARO 14-13638.1-A-E		13. NUMBER OF PAGES 116
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES The findings of this report are not to be considered as an official Department of the Army position, unless so designated by other authorized documents.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Short Range Guided Rockets Mallaunch Factors Digital Simulation Single-Axis Control Missile Flight Dynamics Stability and Control of Rolling Missiles Missile Guidance and Control		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report deals with the problem of digitally simulating the motion of a Short Range Air Defense System (SHORADS) rocket during launch and until the rocket has been acquired by a launcher-fixed tracking radar. Physical and mathematical models of the launcher, the rocket per se and its environment, the rocket's control system and the pre-radar-acquisition guidance system are presented. The pre-acquisition guidance sensor is a dual field of view infra- red angle sensor. The rocket is of the folding-wing, tube-launched, boost-		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

442 958

sustain type and rolls at an essentially constant rate. Control torques are generated by deflecting the thrust of the solid-propellant, sustainer rocket motor and are available about a single rocket-fixed axis. The models for the launcher and rocket include the anomalistic effects of tipoff, launcher slew rate, rocket dynamic unbalance and thrust misalignment, aerodynamic forces and moments, jet damping, variable mass, and moments of inertia of the rocket and winds. Two computer codes - one for the launch phase and one for the post-launch, controlled phase - are also described and typical simulation results are presented and discussed. The rocket simulated performs well when various combinations of anomalies are present, but large slew rates of the launcher result in failure of the rocket to enter the field of view of the infrared sensor, if it is launched from a "lagging" tube.

SIMULATION OF THE FLIGHT OF A SHORT RANGE
AIR DEFENSE SYSTEM ROCKET BEFORE
RADAR ACQUISITION

FINAL REPORT

by

John E. Cochran, Jr.
and
James R. Beaty

U.S. ARMY RESEARCH OFFICE

Grant DAAG29-76-G-0069

Administered through
ENGINEERING EXPERIMENT STATION
Auburn University
Auburn, Alabama 36830

APPROVED FOR PUBLIC RELEASE:
DISTRIBUTION UNLIMITED

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DDC	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	4-7-76 11/7/76 SPECIAL
A	

April 1977

PREFACE

This report describes physical and mathematical models which form the basis for two digital computer codes which can be used to simulate the motion of a Short Range Air Defense System (SHORADS) rocket before it is acquired by a launcher fixed tracking radar. Listings of the two computer codes as well as typical simulation results are also included herein.

Much of the material in this report also constitutes part of Mr. James R. Beaty's Master of Science thesis.

The assistance of Mr. John Howerton of the Army Missile Research and Development Command, Redstone Arsenal, Alabama, who provided technical data for this study is gratefully acknowledged.

Dr. Joe W. Reece, Co-Project Leader, is presently on leave from Auburn University and did not participate in the writing of this report. However, his contributions to the overall effort during its initial stages were indispensable in regard to the successful completion of this study.

The authors also gratefully acknowledge the patience and expertise of Mrs. Marjorie McGee, who typed the manuscript.



John E. Cochran, Jr.
Co-Project Leader

TABLE OF CONTENTS

PREFACE	1
LIST OF FIGURES	iv
LIST OF TABLES	vi
SECTION 1. INTRODUCTION	1
1.1 General Comments	1
1.2 Scope of the Present Study	1
SECTION 2. LAUNCH DYNAMICS	5
2.1 Introductory Comments	5
2.2 Launcher/Rocket System Physical Model	5
2.3 Launcher/Rocket System Mathematical Model	6
2.4 Initial Conditions for the Flight Phase	13
SECTION 3. FLIGHT DYNAMICS	17
3.1 Introductory Comments	17
3.2 Physical Model for the Rocket	17
3.3 Equations of Motion	19
SECTION 4. GUIDANCE AND CONTROL	32
4.1 General Comments	32
4.2 Guidance	32
4.3 Control	36
SECTION 5. SIMULATION RESULTS	43
5.1 General Comments	43
5.2 Data Used in the Simulations	44
5.3 Simulation Results	44

TABLE OF CONTENTS (CONT)

SECTION 6. SUMMARY, CONCLUSIONS AND SUGGESTIONS	78
6.1 Summary	78
6.2 Conclusions	78
6.3 Suggestions	79
REFERENCES	80
APPENDIX A. LISTING OF MISSILE SIMULATION (MISSIM) PROGRAM . . .	81
APPENDIX B. LISTING OF SIMPLIFIED MISSILE/LAUNCHER (MISSLNCH) SIMULATION PROGRAM	98
APPENDIX C. AERODYNAMIC CHARACTERISTICS	106

LIST OF FIGURES

<u>Figure No.</u>		<u>Page No.</u>
2-1	Launcher Model	6
2-2	Geometry During Tipoff	9
2-3	Orientation of the Rocket-Fixed System	14
2-4	Geometry at the End of Tipoff.	15
3-1	Solid and Fluid Portions of the Rocket	18
3-2	Rocket with Wings Unfolded	18
3-3	Thrust Misalignment Angles	21
3-4	Control Force Orientation	23
3-5	Aerodynamic Forces and Moments	23
3-6	Notation Used in Equations of Motion	28
3-7	Geometry of Solid Propellant Charge	28
4-1	IR Guidance Geometry	33
4-2	Target Location	35
4-3	Control Regions	38
4-4	Non-rolling Coordinate System	38
5-1	Pitch Control Time, Δt_y , for Simulation 1	49
5-2	Commanded Non-rolling Angular Rate, Q_{NR_c} , for Simulation 1	50
5-3	Non-rolling Angular Rate, Q_{NR} , for Simulation 1	51
5-4	Rocket-fixed Angular Rate, Q , for Simulation 1	52
5-5	Yaw Control Time, Δt_y , for Simulation 1	53
5-6	Commanded Non-rolling Angular Rate, R_{NR_c} , for Simulation 1	54
5-7	Non-rolling Angular Rate, R_{NR} , for Simulation 1.	55
5-8	Rocket-Fixed Angular Rate, R , for Simulation 1	56

LIST OF FIGURES (CONT)

<u>Figure No.</u>		<u>Page No.</u>
5-9	Angle of Attack Time History	58
5-10	Sideslip Angle Time History	59
5-11	Pitch Angle Time History	60
5-12	Yaw Angle Time History	61
5-13	Yaw Angle vs. Pitch Angle	63
5-14	Time History of x_G -Coordinate of C	64
5-15	Trajectory for Simulation 1	65
5-16	Trajectory for Simulation 2	66
5-17	Trajectory for Simulation 3	68
5-18	Trajectory for Simulation 4	69
5-19	Trajectory for Simulation 5	71
5-20	Trajectory for Simulation 6	73
5-21	Trajectory for Simulation 7	74
5-22	Trajectory for Simulation 8	75
5-23	Trajectory for Simulation 9	76
5-24	Trajectory for Simulation 10	77

LIST OF TABLES

Table No.

5-1	Launch Phase Data	45
5-2	Description of Flight Conditions	46
5-3	Flight Phase Data	47
C-1	Aerodynamic Characteristics	107

SECTION 1. INTRODUCTION

1.1 General Comments

The rising costs of design, development and testing of modern weapon systems coupled with the availability of high-speed digital computers have led to increased utilization of digital computer codes to simulate proposed and existing weapon systems. A wealth of information about the operation of a weapon system can be obtained from such simulation codes if they are properly devised. The type of information may range from results which are only order-of-magnitude estimates of say, accelerations and angular rates, to very accurate predictions or reproductions of the system behavior as a function of time. This report describes the development and use of two simulation codes of moderate complexity which are intended for use in determining the effects of various anomalies, such as dynamic imbalance and winds, and of certain inherent characteristics of a particular type of weapon system.

1.2 Scope of the Present Study

This report deals with the problem of simulating the motion of a Short Range Air Defense System (SHORADS) type rocket during the "capture" or pre-radar-acquisition portion of flight. This portion of the rocket's flight includes its launch from a tube attached to the rotating turret of a ground-based mobile launcher and the rocket's motion prior to its being acquired by a launcher-based tracking radar. The part of the flight after launch and prior to radar acquisition can be divided into two subparts. The first subpart is a stabilized, but unguided, flight phase which begins when the

aft end of the rocket leaves the launch tube and ends when the aft end of the rocket enters the field of view (FOV) of an infrared (IR) angle sensor, the sight axis of which is bore-sighted with that of the radar. (The IR sensor senses the position, within its conical FOV, of flares on the rocket's aft end.) The second subpart is a flight phase in which signals from the IR sensor and pre-specified range versus time relationships are used to determine guidance commands for the rocket. This phase begins when the IR sensor acquires the rocket's flares and ends when the tracking radar acquires contact with the rocket.

A simulation code has previously been developed¹ for the purpose of simulating the motion of a SHORADS type rocket from launch tube exit until the end of its flight. That code is very complicated, due to the fact that an attempt was made to simulate the actual operation of all subsystems. It does not, however, contain a provision for simulating the launch phase, nor does it contain a rocket mathematical model which includes certain anomalies considered in this study and included in the "post-launch" simulation code described herein.

Two digital computer codes were developed during this study. The first is a simplified launch dynamics code and the second is the post-launch simulation code cited above.

Section 2 of this report contains a description of the physical and mathematical models which form the basis for the launch dynamics code. The launcher is modeled as a rigid, fixed base to which a rigid turret with two rigid launch tubes is attached. The turret is assumed to rotate about a vertical axis with constant angular speed. For launch dynamics purposes, the rocket is modeled as a constant mass, axisymmetric rigid body. The

effects of tipoff of the rocket are modeled. The desired output of the launch dynamics code is the initial state vector of the rocket at the instant its aft end leaves the launch tube.

The physical and mathematical models of the rocket are presented in Section 3. The rocket is modeled as a variable mass body in the sense that the indirect effects of variations in its mass and moments of inertia with time and the direct effects of thrust, torque due to thrust, and jet damping due to internal flow are modeled. The rocket modeled herein is of the folding-wing, cruciform, boost-sustain type. It is assumed to be spun-up to a relatively low roll rate (say, 5 cps) during launch and to maintain an essentially constant roll rate after launch. The thrust of the rocket as a function of time is assumed to be available in tabular form. Its mass is modeled as a piecewise linear function of time. Also, the aerodynamics characteristics of the rocket are assumed to be available in tabular form. For example, C_N , the normal force coefficient, is assumed to be available as a tabulated function of the flight Mach number and the incidence angle of the rocket.

Control is accomplished by deflecting the thrust vector of the sustainer rocket motor with small "control vanes" which are located at the exit plane of the sustainer nozzle. Control torques about a single transverse axis fixed in the rocket are generated by the deflected thrust. Since the rocket is rolling rapidly, control torques can be applied alternately to control pitch and yaw plane motions of the rocket. The magnitude of the control torque produced by deflecting the control vanes is assumed constant (a delay time is, however, modeled) and the "amount" of control is determined by the length of time the vanes are deflected in a given direction; i.e., the dwell time.

A simple control law is described in Section 4. Under this law, the dwell time is determined as a linear combination of deviations of the rocket's center of mass from the launcher-to-target line-of-sight, the time rates of change of these deviations and the pitch rate of the rocket. The choice of control gains which appear in the control law is also discussed in Section 4.

Results obtained using the launch dynamics (LD) and pre-radar-acquisition (PRA) codes are presented in Section 5. Plots of the flight paths of the rocket for when its motion is affected by various anomalistic factors are shown and discussed.

Section 6 contains a summary of the main findings of this study and also contains some recommendations for further work in simulating the launch of SHORADS rockets.

Throughout this report, the words rocket and missile will be used interchangeably.

SECTION 2. LAUNCH DYNAMICS

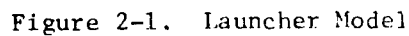
2.1 Introductory Comments

Often, the launch phase is neglected when simulations of rocket systems are conducted. The reason for this neglect often is a belief on the part of the one conducting the simulations that the effects of the rocket's motion during the launch phase may be accounted for by varying the initial conditions for the post-launch motion. This may be the case, but one should not be led to believe that it is always so. The main fault of the process of simply varying initial conditions to simulate "bad launches," tipoff, etc., is that the initial conditions may not be consistent and hence may lead to spurious results.

The launch phase is not as adequately treated herein as it could have been if the majority of the time available had not been spent on the post-launch phase. At the beginning of this study, it was anticipated that an existing launch dynamics code² would be used to simulate the launch phase. That code did not appear suitable, however, since it does not contain a model for a launcher with a rapidly rotating turret. Rather than modify it, a relatively simple physical model for the launcher/rocket system during launch was formulated, the associated mathematical model derived and a new code incorporating this new model written.

2.2 Launcher/Rocket System Physical Model

The physical model for the launcher/rocket system during the launch phase is shown schematically in Fig. 2-1. It consists of a rigid, fixed base, a rigid turret and two rigid launch tubes. The turret rotates relative to the base about a vertical axis. The launch tubes may be rotated to different



2.3 Launcher/Rocket System Mathematical Model

The launch phase is divided into two parts for the purposes of deriving equations of motion for the rocket (No equations of motion are needed for the launcher, since its motion is specified a priori.). The first part of the launch phase is one in which the rocket translates along a tube, while it spins within the tube. The second part is one in which the rocket is

supported only at a point near its aft end and hence "tips off"; i.e., rotates relative to the tube about a transverse axis.

Motion During the First Part of the Launch Phase

During the constrained translational part of the launch phase, the position of the rocket's center of mass C is given by

$$\underline{R}_C = \underline{R}_A + \underline{r}_{C/A} \quad (2-1)$$

where \underline{R}_A is a vector from point T, the origin of the turret-fixed $Tx_T y_T z_T$ system, to point A, the origin of the tube-fixed $Ax_t y_t z_t$ coordinate system and $\underline{r}_{C/A}$ is a vector from A to C, the center of mass of the rocket. If it is assumed that the turret rotates with a constant angular velocity $\underline{\omega}_T = \omega_T \hat{k}_T$, where \hat{k}_T is a unit vector fixed to the z_T -axis, the acceleration of C is

$$\begin{aligned} \ddot{\underline{R}}_C = & \underline{\omega}_T \times [\underline{\omega}_T \times (\underline{R}_A + \underline{r}_{C/A})] \\ & + 2 \underline{\omega}_T \times \dot{\underline{r}}_{C/A} + \ddot{\underline{r}}_{C/A}, \end{aligned} \quad (2-2)$$

where the small circle denotes differentiation of the components of a vector expressed in terms of the tube-fixed unit vectors \hat{i}_t , \hat{j}_t and \hat{k}_t . Since the z_t - and y_t -motions of the rocket are constrained to be zero during this part of the launch phase, only the x_t -component of $\ddot{\underline{R}}_C$ is needed. By determining this component and also the component of the gravitational force, we find that the equation for the rocket's translation within the tube is

$$\ddot{x}_t = -g \sin \delta_G + x_t \omega_T^2 \cos^2 \delta_G + F_T/m \quad (2-3)$$

where g is the acceleration of gravity, δ_G is the elevation angle of the tube, F_T is the magnitude of the thrust force (friction is neglected) and m is the mass of the rocket.

The rocket also spins within the tube. We let p denote the spin, or roll, rate of the rocket relative to the tube, T_x denote the x_t -component (also the x -component, where x is the longitudinal axis of the rocket) of the torque on the rocket and I_x denotes the rocket's moment of inertia about the x_t -axis. The equation for rotational motion of the rocket relative to the tube is then simply

$$\dot{p} = T_x / I_x . \quad (2-4)$$

The roll angle of the rocket relative to the tube may be found from the kinematic equation

$$\dot{\Delta\phi} = p . \quad (2-5)$$

Tipoff Equations of Motion

For the purposes of this study it is assumed that, when the rocket has traveled a specified distance down the tube, the transverse rotation of the rocket is no longer constrained by the tube and any internal guidance mechanism, such as a carriage. During the interval of time beginning when transverse rotation of the rocket is not constrained and ending when the aft end of the rocket exits the tube, "tipoff" occurs. In formulating the equations of motion for this part of the launch phase, it is assumed that the rocket may rotate transversely about a point a on the circumference of the rocket, which point a is located at the end of the tube (see Fig. 2-2).

We let $\Delta\psi$ and $\Delta\theta$ denote Euler angles which along with $\Delta\phi$ define the relative orientation of the rocket-fixed $Cxyz$ system with respect to the tube-fixed $Ax_t y_t z_t$ system, p , q and r denote the rocket-fixed components of the relative angular velocity, $\omega_{r/t}$, of the rocket with respect to the tube-

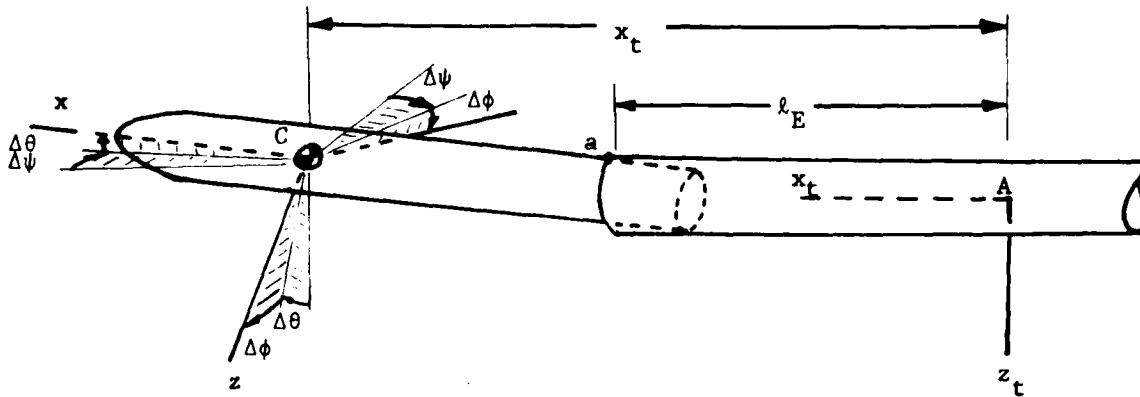


Figure 2-2. Geometry During Tipoff.

fixed system and \underline{I}_r denote the rocket's inertia matrix. Furthermore, we define d as the diameter of the rocket at the point a and assume that d is constant. We also let l_E denote the distance from point A to the fore end of the tube. Finally, we define the following transformation matrices (assuming $\Delta\theta$ and $\Delta\psi$, but not $\Delta\phi$, are small angles):

$$\underline{A}_{r/t} = \begin{bmatrix} 1 & \Delta\psi & -\Delta\theta \\ -\Delta\psi \cos \Delta\phi + \Delta\theta \sin \Delta\phi & \cos \Delta\phi & \sin \Delta\phi \\ \Delta\psi \sin \Delta\phi + \Delta\theta \cos \Delta\phi & -\sin \Delta\phi & \cos \Delta\phi \end{bmatrix} \quad (2-6)$$

$$\underline{C} = \begin{bmatrix} 1 & \Delta\theta \sin \Delta\phi & \Delta\phi \cos \Delta\phi \\ 0 & \cos \Delta\phi & -\sin \Delta\phi \\ 0 & \sin \Delta\phi & \cos \Delta\phi \end{bmatrix} \quad (2-7)$$

Here $\underline{A}_{r/t}$ is the matrix which may be used to transform the x_t -, y_t - and z_t -components of a vector quantity into x -, y -, and z -components and \underline{C} is the matrix such that

$$\begin{bmatrix} \dot{\Delta\phi} \\ \dot{\Delta\theta} \\ \dot{\Delta\psi} \end{bmatrix} = \underline{\underline{C}} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2-8)$$

The methods of Ref. 2 may then be used to obtain the equation of relative rotational motion of the rocket. This equation may be expressed in the matrix form,

$$\begin{aligned} \frac{\circ}{r} \frac{\omega}{r/t} &= \underline{\underline{L}}^{-1} \left\{ \underline{\underline{I}}_r \frac{\tilde{\omega}}{r} r/t \underline{\underline{A}}_{r/t} \frac{\omega}{t} - \frac{\tilde{\omega}}{r} \underline{\underline{I}}_r \frac{\omega}{r} \right. \\ &- m \frac{\tilde{\ell}}{a} \underline{\underline{A}}_{r/t} \underline{\underline{E}}_{23} \left[\left[\underline{\underline{A}}_{t/r} \frac{\tilde{\omega}}{r/t} + \frac{\tilde{\omega}}{t} \underline{\underline{I}}_T \right] \frac{\ell}{a} + 2 \frac{\tilde{\omega}}{t} \underline{\underline{A}}_{t/r} \frac{\tilde{\omega}}{r/t} \frac{\ell}{a} \right. \\ &\left. \left. + \underline{\underline{A}}_{t/T} \frac{\tilde{\omega}}{T} \underline{\underline{I}}_T \left[\frac{R}{T} \underline{\underline{A}} + \underline{\underline{A}}_{T/t} \frac{r}{t} \underline{\underline{C}}/A \right] \right] + \frac{\tilde{\ell}}{a} \underline{\underline{A}}_{r/t} \underline{\underline{E}}_{23} \frac{F^{ex}}{t} r + \underline{\underline{A}}_{r/t} \frac{d}{t} \underline{\underline{A}}_{t/R} \frac{F}{r} T + \frac{T}{r} s \right\}, \end{aligned} \quad (2-9)$$

where the letter r beneath a matrix indicates the x-, y- and z-components of the corresponding vector quantity are used to form the matrix and the following additional definitions have been introduced:[†]

$$\frac{\circ}{r} \frac{\omega}{r/t} = [\dot{p} \quad \dot{q} \quad \dot{r}]^T$$

$$\underline{\underline{I}}_r = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_T & 0 \\ 0 & 0 & I_T \end{bmatrix}$$

$$\underline{\underline{L}} = \underline{\underline{I}}_r - m \frac{\tilde{\ell}}{a} \underline{\underline{A}}_{r/t} \underline{\underline{E}}_{23} \underline{\underline{A}}_{t/r} \frac{\tilde{\ell}}{a}$$

$$\frac{\ell}{a} = [(\underline{x}_t - \underline{\ell}_E) \quad 0 \quad 0]^T$$

[†]A superscript T denotes the matrix transpose.

$$\underline{A}_{t/T} = \begin{bmatrix} \cos \delta_G & 0 & -\sin \delta_G \\ 0 & 1 & 0 \\ \sin \delta_G & 0 & \cos \delta_G \end{bmatrix}$$

$$\underline{A}_{T/t} = \underline{A}_{t/T}^T$$

$$\underline{F}_T = [F_T \quad 0 \quad 0]^T$$

$$\underline{T}_S = [T_x \quad 0 \quad 0]^T$$

$$\underline{\omega}_T = [0 \quad 0 \quad \omega_T]^T$$

$$\underline{\omega}_t = \underline{A}_{t/T} \underline{\omega}_T$$

$$\underline{\omega}_r = \underline{\omega}_r/t + \underline{A}_{r/t} \underline{\omega}_t$$

$$\underline{R}_A = [x_{T_A} \quad y_{T_A} \quad 0]^T$$

$$\underline{d}_t = [d/2 \quad 0 \quad 0]^T$$

$$\underline{\dot{\ell}}_a = [\dot{x}_t \quad 0 \quad 0]^T$$

$$\underline{E}_{23} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\underline{E}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\frac{F_r^{ex}}{t} = [-mg \sin \delta_G \quad 0 \quad mg \cos \delta_G]^T + \underline{A}_{t/r} \frac{F_T}{r}$$

$$\frac{F_T}{r} = [F_T \quad 0 \quad 0]^T$$

$$\frac{T}{r_s} = [T_x \quad 0 \quad 0]^T$$

Also, in Eq. (2-9) and throughout this report, a tilde (\sim) above a 3 x 1 matrix denotes a particular 3 x 3 skew-symmetric matrix formed from the elements of the 3 x 1 matrix. For example,

$$\tilde{\omega}_{rr/t} = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}.$$

The equation which governs the translational motion of the rocket's center of mass may also be obtained by using the methods of Ref. 2. It is

$$\begin{bmatrix} \ddot{x}_t \\ 0 \\ 0 \end{bmatrix} = \underline{E}_1 \underline{A}_{t/T} \tilde{\omega}_T \overset{\circ}{r}_t C/A + \underline{A}_{t/T} \tilde{\omega}_T \tilde{\omega}_T \left[\frac{R}{T} A + \underline{A}_{T/t} \frac{r}{t} C/A + \frac{1}{m} \frac{F_r^{ex}}{t} \right], \quad (2-10)$$

where

$$\overset{\circ}{r}_t C/A = \underline{A}_{t/r} \tilde{\omega}_{r/t} \frac{l}{r} a + \underline{A}_{t/r} \overset{\circ}{l} a.$$

Equations (2-9) and (2-10) include the effects of the constraint forces which cause the point a to move with the turret. They, along with Eq. (2-8), comprise the equations of motion for the tipoff part of the launch phase.

2.4 Initial Conditions for the Flight Phase

The main purpose of the launch phase equations is to provide realistic initial conditions for the flight phase; i.e., that phase beginning when the rocket is free of the launcher's constraints. At the end of the launch phase simulation in which the requisite differential equations are integrated numerically, we have the values of $x_t = l_E +$ the distance from the aft end of the rocket to the center of mass, $\Delta\phi$, $\Delta\theta$, $\Delta\psi$, p , q , r and $\lambda_G = \omega_T(t_{\text{exit}} - t_0)$, where t_{exit} is the time the launch phase ends and t_0 is the time the rocket starts to move.

The initial conditions we need for the flight phase is the following: the x_G -, y_G - and z_G -coordinates of the rocket's center of mass C; the Euler angles ψ , θ and ϕ which are used to define the orientation of the rocket-fixed coordinate system Cxyz with respect to an earth-fixed system OXYZ; the inertial components of the velocity of C, U, V and W; P, Q and R, the x-, y-, and z-components, respectively, of the angular velocity of the Cxyz system relative to the OXYZ system. We shall use a small e as a subscript here to denote the value at the instant of exit. Also, the vector r_G , from point G to point A is introduced at this point. Using previous results and resorting to Figures 2-3 and 2-4, we obtain first

$$\begin{pmatrix} P & Q & R \end{pmatrix}_e^T = \begin{pmatrix} p & q & r \end{pmatrix}^T + \frac{A}{r/T} \omega_T \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}_e, \quad (2-11)$$

then

$$\begin{pmatrix} U & V & W \end{pmatrix}_e^T = \frac{A}{r/T} \frac{\omega_T}{T} \frac{R}{T} A + \frac{A}{r/t} \frac{\dot{r}_C}{t} A \quad (2-12)$$

and

$$\begin{pmatrix} x_G & y_G & z_G \end{pmatrix}_e^T = \frac{A}{r/t/T} \left(\frac{R}{T} A - \frac{r_G}{T} \right) + \frac{r_C}{t} A. \quad (2-13)$$

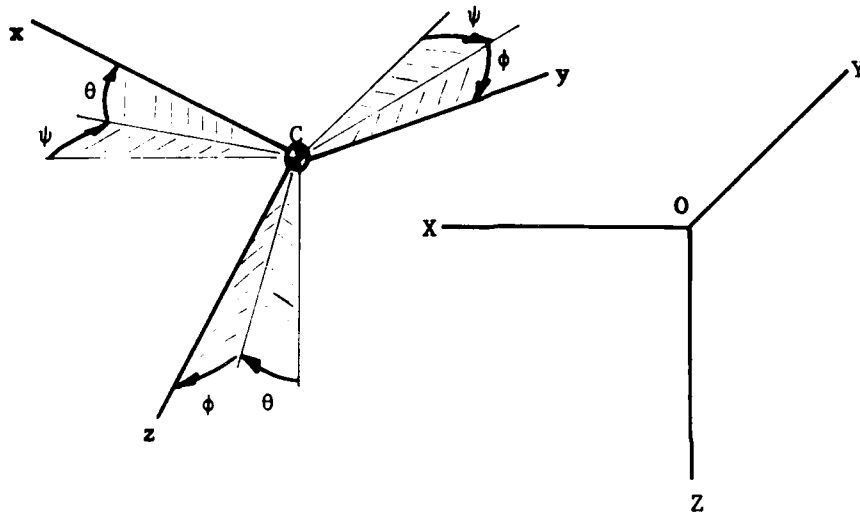


Figure 2-3. Orientation of the Rocket-Fixed System.

To determine the initial values of the Euler angles ψ , θ and ϕ , we construct the direction cosine matrix for the $G \ x_G \ y_G \ z_G$ system (the same matrix as that for the $A x_t y_t z_t$ system) using first the angles λ_G , δ_G , $\Delta\psi$, $\Delta\theta$, $\Delta\phi$. This matrix we call $L_{b/I}$, where b/I denotes that the rotation is from the inertial OXYZ system to the body-fixed system Cxyz. This matrix can also be constructed by using the Euler angles ψ , θ and ϕ . By comparing terms in the two different expressions for $L_{b/I}$, we therefore find that

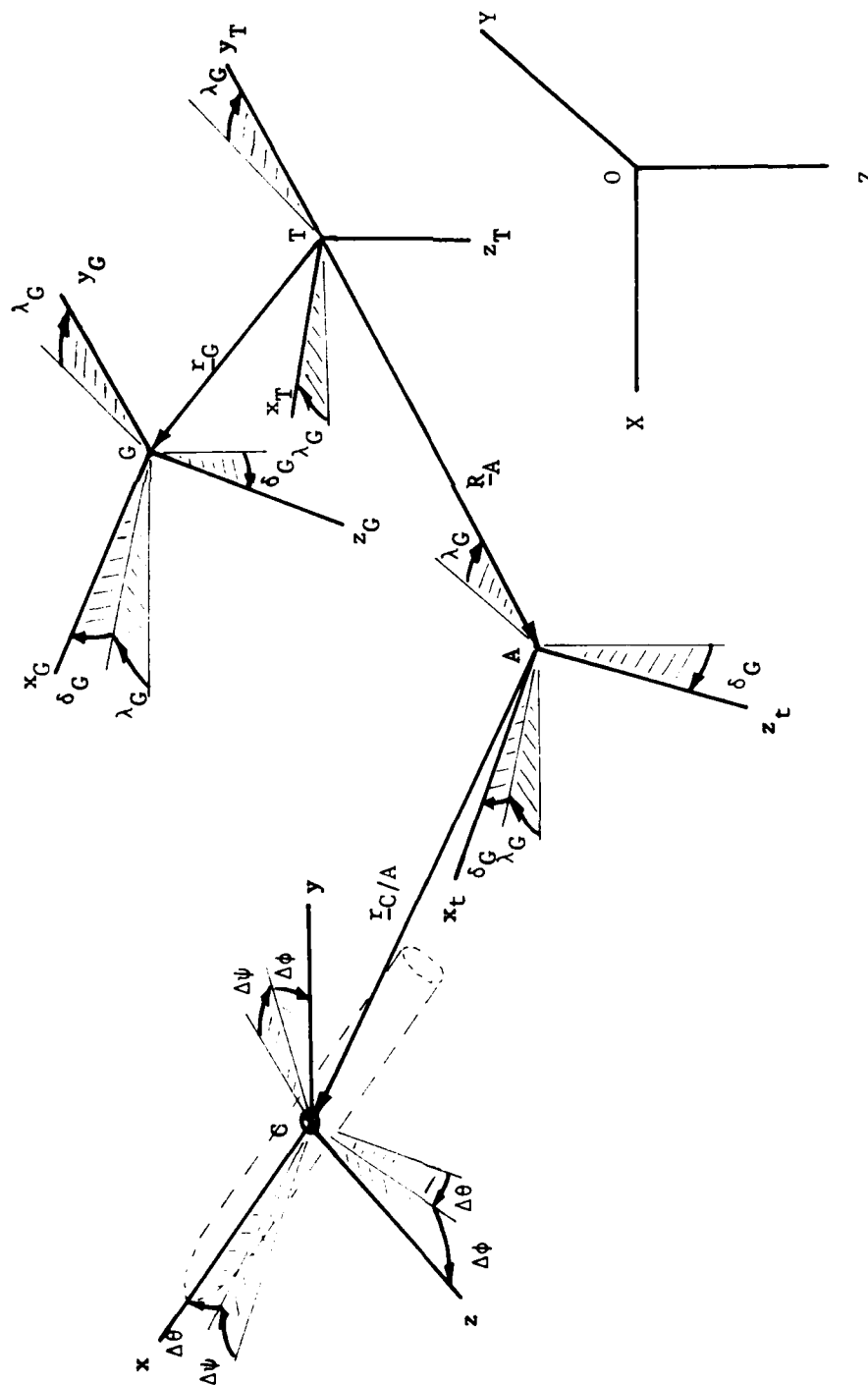


Figure 2-4. Geometry at the End of Tipoff.

$$\tan \psi_e = \left[\frac{\cos \delta_G \sin \lambda_G + \Delta\psi \cos \lambda_G - \Delta\theta \sin \delta_G \sin \lambda_G}{\cos \delta_G \cos \lambda_G - \Delta\psi \sin \lambda_G - \Delta\theta \sin \delta_G \cos \lambda_G} \right]_e \quad (2-14a)$$

$$\tan \phi_e = \left[\frac{\sin \delta_G [\Delta\psi \cos \Delta\phi - \Delta\theta \sin \Delta\phi] + \cos \delta_G \sin \Delta\phi}{-\sin \delta_G [\Delta\psi \sin \Delta\phi + \Delta\theta \cos \Delta\phi] + \cos \delta_G \cos \Delta\phi} \right]_e \quad (2-14b)$$

$$\sin \theta_e = [\sin \delta_G + \Delta\theta \sin \delta_G]_e \quad (2-14c)$$

We have, at this point, developed the basis for simulating the launch phase and have also obtained equations from which we can determine initial conditions for the flight phase, given the final conditions for the launch phase. In the next section, we develop the flight phase model.

SECTION 3. FLIGHT DYNAMICS

3.1 Introductory Comments

In this section, we develop the mathematical model for the rocket. This model is essentially the same as that developed in Ref. 2 and used there as the basis for the simulation of the free-flight phase of a "free," or uncontrolled, rocket. The differences in the two models are that (1) additional forces and moments due to the deflection of the sustainer motor's thrust vector are included here and (2) the time variations of the mass of the rocket and of its centroidal moments of inertia are assumed, for the purposes of this study to be prescribed a priori as explicit functions of time.

3.2 Physical Model for the Rocket

The rocket is modeled physically as a variable mass body composed of a rigid portion and a fluid portion (see Fig. 3-1). The rigid portion consists of the unexpendable parts of the rocket, plus the part of the solid propellant not yet expended at time t . The fluid portion consists of that part of the solid propellant which has been burned (or is burning) and is flowing as a fluid within the rocket, having not yet passed through the rocket nozzles. For the purposes of modeling the internal flow, the geometry of the rocket is simplified by assuming that the flow is within the core of a cylindrical, solid propellant charge. Since the internal flow model is actually used only in estimating the jet-damping torque, this simple model is considered sufficient.

The rocket, apart from the solid propellant, consists (see Fig. 3-2) first of all a shell, or fuselage, which contains the warhead, the solid

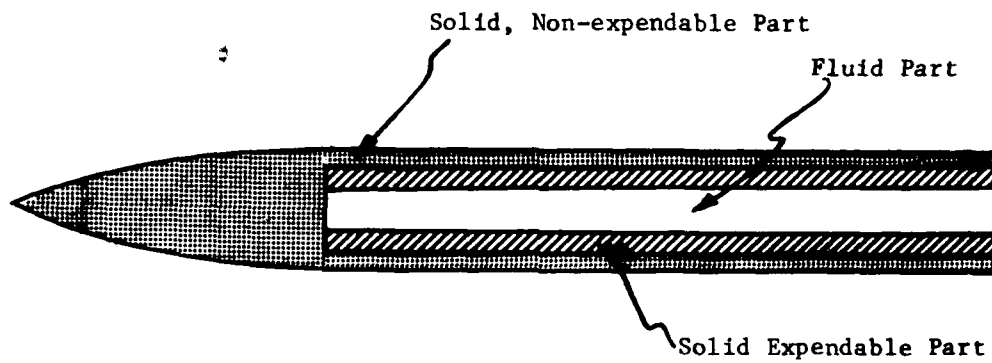
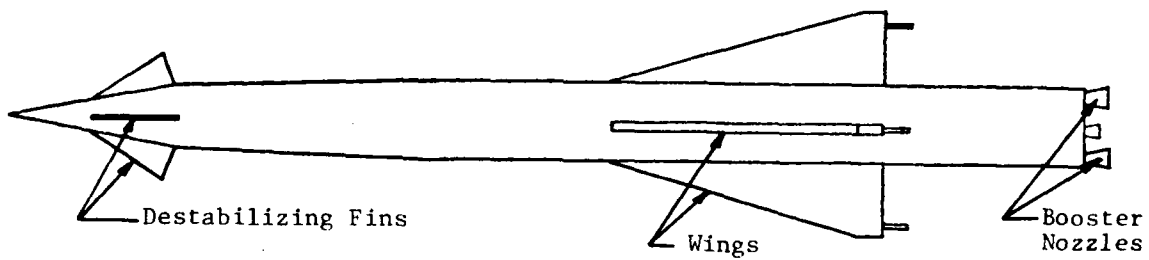
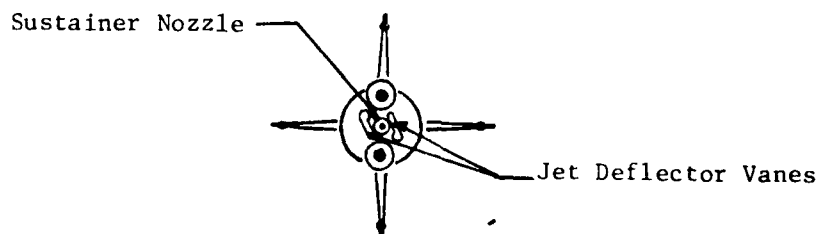


Figure 3-1. Solid and Fluid Portions of the Rocket.



(a) Side View



(b) Rear View

Figure 3-2. Rocket with Wings Unfolded,

propellant, the rocket nozzles, a pitch rate gyro, a roll angle gyro, the actuating mechanism for the thrust deflector vanes, the vanes themselves, the electronic equipment needed to receive and process guidance commands and a device which controls the motion of destabilizing fins. The last six items listed will be discussed more fully in the section on guidance and control (see Section 4). Four wings are attached to the rocket's fuselage in a cruciform configuration. These wings fold out as the rocket exits the launch tube. Also, there are four small fins which are extended through the slots near the nose of the rocket. They are the destabilizing fins mentioned above. They are extended in a pre-programmed manner so that the aerodynamic center of the rocket moves forward as the center of mass moves forward (due to propellant consumption), with the result that a fairly constant static margin is maintained during the rocket's powered flight.

3.3 Equations of Motion

The equations of motion for the rocket may be obtained using the procedure followed in Ref. 1, or, for example, that of Ref. 3.

Translational Motion

The equation which governs the translation of the center of mass of the rocket may be written in the form,

$$m \ddot{\underline{V}}_C = \underline{F}_T + \underline{F}_C + \underline{F}_{\text{Coriolis}} + \underline{F}_{\text{unsteady}} + \underline{F}_{\text{ex}}, \quad (3-1)$$

where \underline{F}_T is the thrust "force"[†] (see Ref. 4) which is generated by the outflow of the fluid part of the rocket, \underline{F}_C is the control force (also generated by the same mechanism as \underline{F}_T , but segregated here for convenience). $\underline{F}_{\text{Coriolis}}$

[†]The word "force" is enclosed in quotation marks here to indicate that it pertains to a term in the equation of motion which is treated as a force, but actually arises due to the fact that mass is being expelled from the rocket and strictly speaking is a reaction force on the solid part of the rocket due

is the "force" required to accelerate the mass of fluid within the rocket transversely, $\underline{F}_{\text{unsteady}}$ is the "force" resulting from the time rate of change of fluid flow properties at a given spatial position and $\underline{F}_{\text{ex}}$ includes all forces on the rocket due to external influences such as the atmosphere and gravity.

In most cases, $\underline{F}_{\text{Coriolis}}$ and $\underline{F}_{\text{unsteady}}$ are not significant. Hence, we will neglect these two "forces." Then, by writing

$$\underline{V}_C = U \hat{i} + V \hat{j} + W \hat{k}, \quad (3-2)$$

where \hat{i} , \hat{j} and \hat{k} are unit vectors fixed to the rolling rocket which has angular velocity,

$$\underline{\Omega} = P \hat{i} + Q \hat{j} + R \hat{k}, \quad (3-3)$$

and letting $\dot{\underline{V}}_C = \dot{U} \hat{i} + \dot{V} \hat{j} + \dot{W} \hat{k}$, we may rewrite Eq. (3-1) in the form,

$$\dot{\underline{V}}_C = -\underline{\Omega} \times \underline{V} + \frac{1}{m} [\underline{F}_T + \underline{F}_C + \underline{F}_{\text{ex}}] \quad (3-4)$$

If the rocket is imperfect, which is usually the case, then to account for thrust vector misalignment we write

$$\underline{F}_T = T \cos \alpha_y \cos \alpha_z \hat{i} + T \cos \alpha_y \sin \alpha_z \hat{j} - T \sin \alpha_y \hat{k}, \quad (3-5)$$

where T is the magnitude of the thrust and α_y and α_z are constant angles (as shown in Fig. 3-3) which are measures of the amount and direction of the mechanical thrust misalignment present. It should be pointed out that these angles represent the "equivalent thrust misalignment" resulting from the combined mechanical misalignment of both the booster motor nozzles.

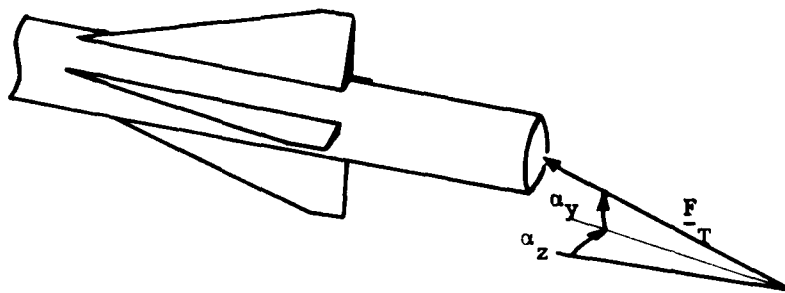


Figure 3-3. Thrust Misalignment Angles.

The control force is discussed more fully in Section 4, but here we note that (see Fig. 3-4).

$$\underline{F}_C = +F[-\sin \delta \hat{j} + \cos \delta \hat{k}], \quad (3-6)$$

where F is the magnitude of the control force and δ is a constant angle representing the rotation of the control plane with respect to the xz -plane.

The external force which acts on the rocket is the resultant of the aerodynamic force and the force due to gravity, hence,

$$\underline{F}_{ex} = \underline{F}_A + \underline{F}_g, \quad (3-7)$$

where

$$\underline{F}_A = -A \hat{i} - Y \hat{j} - N \hat{k} \quad (3-8)$$

and

$$\underline{F}_g = mg[-\sin \theta \hat{i} + \cos \theta \sin \phi \hat{j} + \cos \theta \cos \phi \hat{k}]. \quad (3-9)$$

In Eq. (3-8), A is the axial aerodynamic force component, Y is the y -component of the aerodynamic force and N is the normal aerodynamic force component (see Fig. 3-5). In Eq. (3-9), θ and ϕ are two of the Euler angles which are used to define the orientation of the rocket-fixed coordinate system $Cxyz$ (see Fig. 2-3).

For the purposes of this study, the aerodynamic forces are assumed to be expressible as

$$A = 1/2 \rho S V_R^2 C_A \quad (3-9a)$$

$$Y = 1/2 \rho S V_R^2 C_y \quad (3-9b)$$

and

$$N = 1/2 \rho S V_R^2 C_N \quad (3-9c)$$

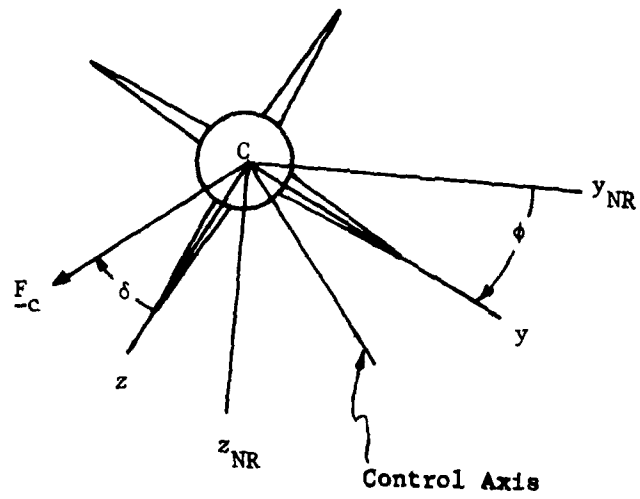


Figure 3-4. Control Force Orientation.

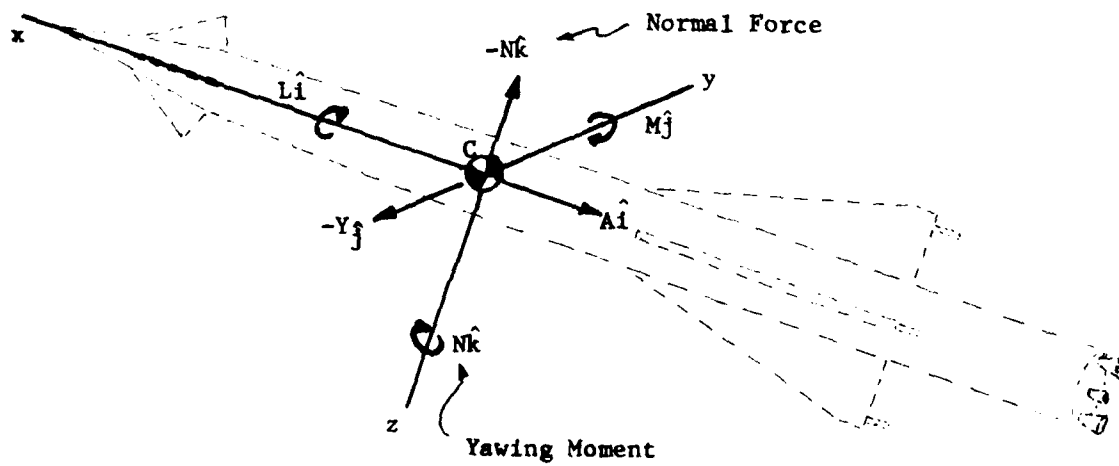


Figure 3-5. Aerodynamic Forces and Moments.

where ρ is the atmospheric density, S is the reference area upon which the aerodynamic coefficients C_A , C_y and C_N are based and V_R is the absolute value of the relative wind that the rocket encounters. Herein

$$V_R = \sqrt{(U_w - U)^2 + (V_w - V)^2 + (W_w - W)^2} \quad (3-10)$$

where U_w , V_w and W_w are the x-, y- and z-components of the velocity of the wind.

The usual assumption that C_A is a function of angle of incidence and Mach number, M , is made and it is further assumed that the incidence angle[†] is small. Then, the following forms for C_A , C_y and C_N are adopted:

$$C_A = C_{A_0}(M) \quad (3-11a)$$

$$C_y = -C_{N_\alpha}(M) \beta \quad (3-11b)$$

$$C_N = C_{N_\alpha}(M) \alpha, \quad (3-11c)$$

where $C_{A_0}(M)$ and $C_{N_\alpha}(M)$ are tabulated functions and $\alpha = \tan^{-1}[(W - W_w)/(U - U_w)]$ and $\beta = \sin^{-1}[(V - V_w)/V_R]$ are the aerodynamic angles of attack and sideslip, respectively.

In the simulation program, the differential equations for V and W are not integrated. Instead, the variables $\alpha_x = \tan^{-1}(W/U)$ and $\beta_x = \sin^{-1}(V/V_c)$ are used in place of V and W for computational purposes. The variables α_x and β_x are the aerodynamic angles of attack and sideslip, respectively if there is no wind. The derivatives \dot{V} and \dot{W} are computed using Eq. (3-4) and $\dot{\alpha}_x$ and $\dot{\beta}_x$ are obtained from

$$\dot{\alpha}_x = (U\dot{W} - W\dot{U})/(U^2 + W^2) \quad (3-12a)$$

[†]The incidence angle referred to here is that angle between the relative

and

$$\dot{\rho}_{\mathbf{x}} = [\dot{V}(U^2 + W^2) - V(U\dot{U} + W\dot{W})] / [V_C^2 \sqrt{U^2 + W^2}] \quad (3-12b)$$

The kinematic equation for translation is obtained by noting that the velocity of the center of mass of the rocket may be written either as

$$\underline{V}_C = \dot{X} \hat{I} + \dot{Y} \hat{J} + \dot{Z} \hat{K}, \quad (3-13)$$

where \hat{I} , \hat{J} and \hat{K} are unit vectors associated with the earth-fixed OXYZ system, or as

$$\underline{V}_C = U \hat{i} + V \hat{j} + W \hat{k}. \quad (3-14)$$

Since the components of \underline{V}_C in the two systems are related by the transformation matrix (see Fig. 2-3) used previously in Section 2,

$$\underline{L}_{b/I} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ -c\phi s\psi & c\phi c\psi & s\phi c\theta \\ +s\phi s\theta c\psi & +s\phi s\theta s\psi & \\ s\phi s\psi & -s\phi c\psi & c\phi c\theta \\ +c\phi s\theta c\psi & +c\phi s\theta s\psi & \end{bmatrix}. \quad (3-15)$$

The required kinematic equation is

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \underline{L}_{I/b} \begin{bmatrix} U \\ V \\ W \end{bmatrix}, \quad (3-16)$$

where $\underline{L}_{I/b} = \underline{L}_{b/I}^T$.

Rotational Motion

The equation governing the rotational motion of the rocket about its center of mass C is

$$\underline{\underline{I}} \dot{\underline{\Omega}} + \underline{\Omega} \times \underline{\underline{I}} \cdot \underline{\Omega} = \underline{T}_T + \underline{T}_C + \underline{T}_{\text{unsteady}} + \underline{T}_{\text{Coriolis}} + \underline{T}_{\text{ex}}, \quad (3-17)$$

where $\underline{\underline{I}}$ is the centroidal inertia dyadic of the rocket, \underline{T}_T is the "torque"[†] about C due to the thrust, \underline{T}_C is the control "torque" about C, $\underline{T}_{\text{unsteady}}$ is the "torque" about C due to the unsteadiness of the flow, $\underline{T}_{\text{Coriolis}}$ is the "torque" about which is due to the time rate of change of the direction of the motion of the fluid which is flowing inside the rocket caused by rotation of the rocket and $\underline{T}_{\text{ex}}$ is the torque about C due to aerodynamic reactions.

The inertia dyadic $\underline{\underline{I}}$ is assumed to be general in form to account for mass imbalance which may be present due to manufacturing imperfections. The matrix counterpart of $\underline{\underline{I}}$ expressed in terms of rocket-fixed elements is

$$\underline{\underline{I}} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}. \quad (3-18)$$

In the ideal case, $I_{xy} = I_{xz} = I_{yz} = 0$ and $I_{yy} = I_{zz}$.

The torque due to thrust is expressed explicitly as

$$\underline{T}_T = (\underline{\ell}_T - \underline{u}) \times \underline{F}_T, \quad (3-19)$$

[†]The word "torque" is enclosed in quotation marks here to indicate that it pertains to a term in the equation of motion which is treated as an external torque, but actually arises from the fact that mass is being expelled from the rocket and strictly speaking is a reaction torque on the solid part of the rocket due to the internal flow or expelled mass. The quotation marks will be dropped in later references to the same term.

where \underline{l}_T is a vector from an arbitrary point in the non-expendable part of the rocket to a point on the line of action of the resultant thrust vector. Also, in Eq. (3-18) \underline{u} is a vector from a to C (see Fig. 3-6).

The control torque, like the control force, will be discussed more fully in Section 4. However, we note here that it has the form,

$$\underline{T}_c = (\underline{r}_T - \underline{u}) \times \underline{F}_c, \quad (3-20)$$

where \underline{r}_T is a vector from point a to the line of action of the control force.

For the purposes of this study the torque due to the unsteadiness of the flow is assumed to be negligible compared to other torques.

The Coriolis torque is the so-called "jet-damping" torque, \underline{T}_{JD} , and is retained. By assuming a cylindrical solid propellant charge and constant mass flow rate per unit length of the charge, an approximate expression for $\underline{T}_{Coriolis}$ may be obtained.² The expression used in this simulation study is

$$\underline{T}_{JD} \equiv \underline{T}_{Coriolis} = -2 C_{JD} [Q \hat{j} + R \hat{k}] \quad (3-21)$$

Here, the jet damping coefficient, C_{JD} , is given by

$$C_{JD} = -\dot{m} [(u_1 - x_2'/3) \ell_p + (x_1')^2/3], \quad (3-22)$$

where \dot{m} is the time rate of change of the rocket's mass, $u_1 = \underline{u} \cdot \hat{i}$, ℓ_p is the length of the propellant charge and x_1' and x_2' are the coordinates of the aft and fore ends, respectively, of the propellant charge as measured from point a as shown in Fig. 3-7.

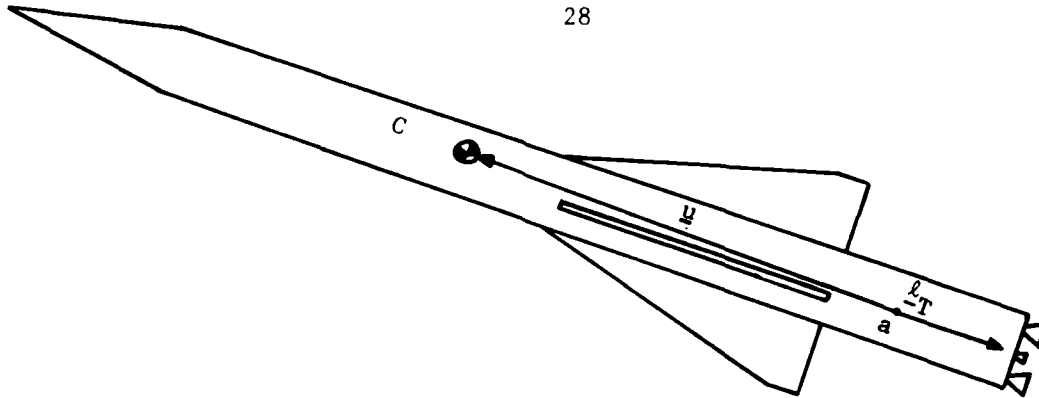


Figure 3-6. Notation used in Equations of Motion.

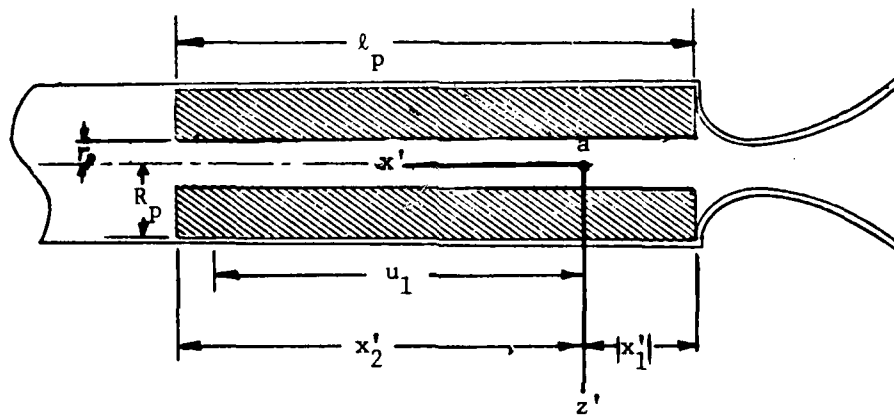


Figure 3-7. Geometry of Solid Propellant Charge.

The external torque about the rocket's center of mass is due to the resultant aerodynamic force. This aerodynamic torque may be expressed in the conventional form,

$$\underline{T}_{\text{ex}} = \underline{T}_A = L \hat{i} + M \hat{j} + N \hat{k} \quad (3-23)$$

where L , M and N^+ are the rolling, pitching and yawing components, respectively, of \underline{T}_A . The components L , M and N may be expressed in the forms,

$$L = 1/2 \rho S V_R^2 d C_\ell \quad (3-24a)$$

$$M = 1/2 \rho S V_R^2 d C_m \quad (3-24b)$$

and

$$N = 1/2 \rho S V_R^2 d C_n, \quad (3-24c)$$

where C_ℓ , C_m and C_n are the usual aerodynamic coefficients and d is the reference length upon which these coefficients are based.

In general,

$$C_\ell = C_\ell(M, \alpha, \beta, P) \quad (3-25a)$$

$$C_m = C_m(M, \alpha, \dot{\alpha}, Q) \quad (3-25b)$$

and

$$C_n = C_n(M, \dot{\beta}, \ddot{\beta}, R), \quad (3-25c)$$

where M is used here to denote the flight Mach number. For small α_A and β_A , which we assume is the case, C_ℓ can be adequately represented by the expression,

$$C_\ell \approx [d/(2V_R)] C_{\ell p}(M) P, \quad (3-26a)$$

⁺ Note that the symbols M and N are used elsewhere for the flight Mach number and the normal component of the aerodynamic force, respectively. This notation is also conventional and it should be clear from the context in which the symbols appear which meaning is to be attached to each of them.

where C_{ℓ_p} is a non-dimensional stability derivative defined as

$$C_{\ell_p} \equiv (2V_R/d)(\partial C_{\ell}/\partial P)_{P=0}.$$

Also, for small α , β , P and Q ,

$$C_m \approx C_{m_\alpha} \alpha + [d/(2V_R)] C_{m_q} Q, \quad (3-26b)$$

where

$$C_{m_\alpha} \equiv (\partial C_m / \partial \alpha) \Big|_{\alpha=0}$$

and

$$C_{m_q} \equiv (\partial C_m / \partial Q)(2V_R/d) \Big|_{Q=0}.$$

Since the rocket is geometrically symmetric, $C_{n_\beta} = -C_{m_\alpha}$ and $C_{n_r} = C_{m_q}$, so that we also write

$$C_n \approx -C_{m_\alpha} \beta + [d/(2V_R)] C_{m_q} R. \quad (3-25c)$$

In this report, the quantities C_{ℓ_p} , C_{m_α} and C_{m_q} are assumed to be given functions of the flight Mach number in the form of tabulated values. The stability derivative C_{m_α} is a function of the static margin; i.e., $(x_{cp} - x_{cg})/d$ where x_{cp} and x_{cg} are the distances from the rocket's nose to its aerodynamic center and its center of mass, respectively. This quantity is a function of M because the distance x_{cp} varies directly with Mach number and x_{cg} varies "indirectly" with Mach number, since as the rocket's fuel is consumed and its center of mass moves forward, the Mach number increases until burnout of the booster motor. The destabilizing fins shown in Fig. 3-2 are extended as the fuel is consumed in such a way that $(x_{cp} - x_{cg})$ remains fairly constant during powered flight; hence, C_{m_α} also remains fairly constant. The

destabilizing fins do not make the rocket unstable, but they only make it less statically stable than it would otherwise be at Mach numbers near and above unity.

The basic mathematical model of the rocket has been presented. We turn, in the next section, to the task of formulating guidance and control algorithm for the system composed of the rocket and the sensing and guidance command generating equipment on the launcher.

SECTION 4. GUIDANCE AND CONTROL

4.1 General Comments

The guidance and control of the rocket prior to radar acquisition are considered in this section. First, the problem of guiding the rocket toward a moving target is addressed. Then, the control algorithm used in this investigation is described.

4.2 Guidance

It is assumed that guidance commands are formulated in the launch vehicle by means of a special purpose guidance computer. The input to the guidance computer is angular data acquired by the launcher-fixed IR sensor. This information is available, however, only if the aft end of the rocket (which contains flares) is within the field of view (FOV) of the IR sensor. The sensor is assumed to have two FOV's - one for acquisition of the rocket and a more narrow one for use after acquisition by the sensor, but prior to radar acquisition (see Fig. 4-1). The guidance computer is assumed to contain a pre-programmed range versus time relation, so that the spherical coordinates, $(R_G, \epsilon_z, \epsilon_y)$ (see Fig. 4-1) where R_G is the range of the rocket and ϵ_z and ϵ_y are the "aximuth" and "elevation" angles, respectively, of the rocket with respect to $Gx_Gy_Gz_G$ coordinate frame. Furthermore, for the purposes of this study, it is assumed that values of \dot{R}_G , $\dot{\epsilon}_z$ and $\dot{\epsilon}_y$ are available. These may be estimated by using the range versus time relationship, the angular measurements and the gimbal rates of the radar (which is assumed to perfectly track the target). For use in the current control algorithm, the linear deviations of the rocket from the line of sight (LOS) from the launcher radar to the target (This LOS is the extended x_G -axis)

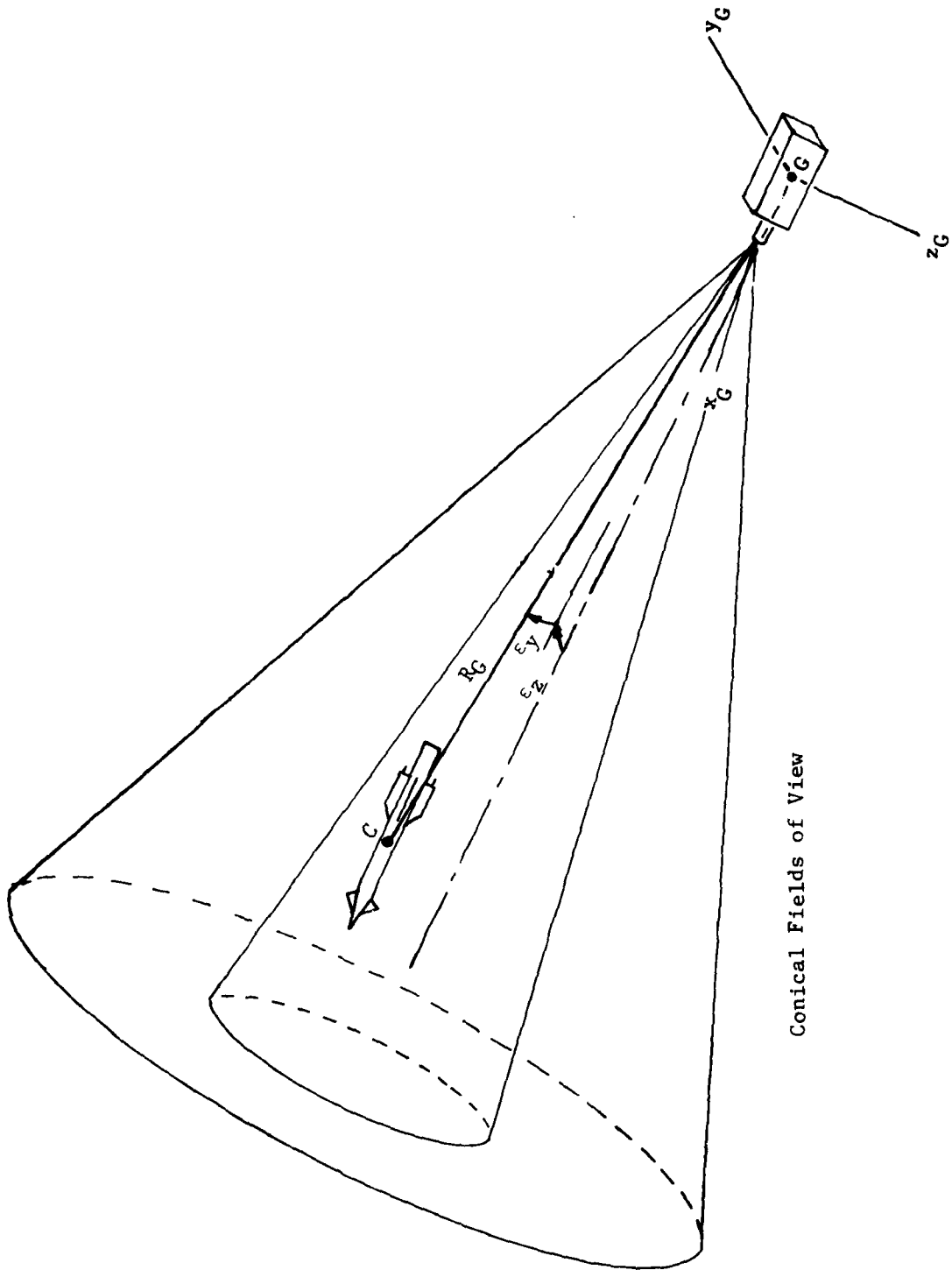


Figure 4-1. IR Guidance Geometry.

are used. These deviations are y_G and z_G which may be computed by using the equations,

$$y_G = R_G \cos \epsilon_y \sin \epsilon_z \quad (4-1a)$$

and

$$z_G = -R_G \sin \epsilon_y \quad (4-1b)$$

The time rates of change of y_G and z_G are also used in the control algorithm. These may be obtained from

$$\begin{aligned} \dot{y}_G = & \dot{R}_G \cos \epsilon_y \sin \epsilon_z - R_G \dot{\epsilon}_y \sin \epsilon_y \sin \epsilon_z \\ & + R_G \dot{\epsilon}_z \cos \epsilon_y \cos \epsilon_z \end{aligned} \quad (4-2a)$$

and

$$\dot{z}_G = -[\dot{R}_G \sin \epsilon_y + R_G \epsilon_y \cos \epsilon_y] \quad (4-2b)$$

In an actual system, these rates could be estimated by using

$$\dot{y}_G \approx -R_G \dot{\lambda}_G \cos \delta_G + \dot{R}_G \epsilon_z \quad (4-3a)$$

and

$$\dot{z}_G \approx -R_G \dot{\delta}_G - \dot{R}_G \epsilon_y \quad (4-3b)$$

for small ϵ_y , ϵ_z , $\dot{\epsilon}_y$, $\dot{\epsilon}_z$ and pre-radar acquisition values of R_G .

In this report, we consider only constant-speed targets (see Fig. 4-2).

For such targets, the angles λ_G and δ_G are given by

$$\lambda_G = \tan^{-1}(y_T/x_T) \quad (4-4a)$$

and

$$\delta_G = \sin^{-1}(-z_T/\rho_T) \quad (4-4b)$$

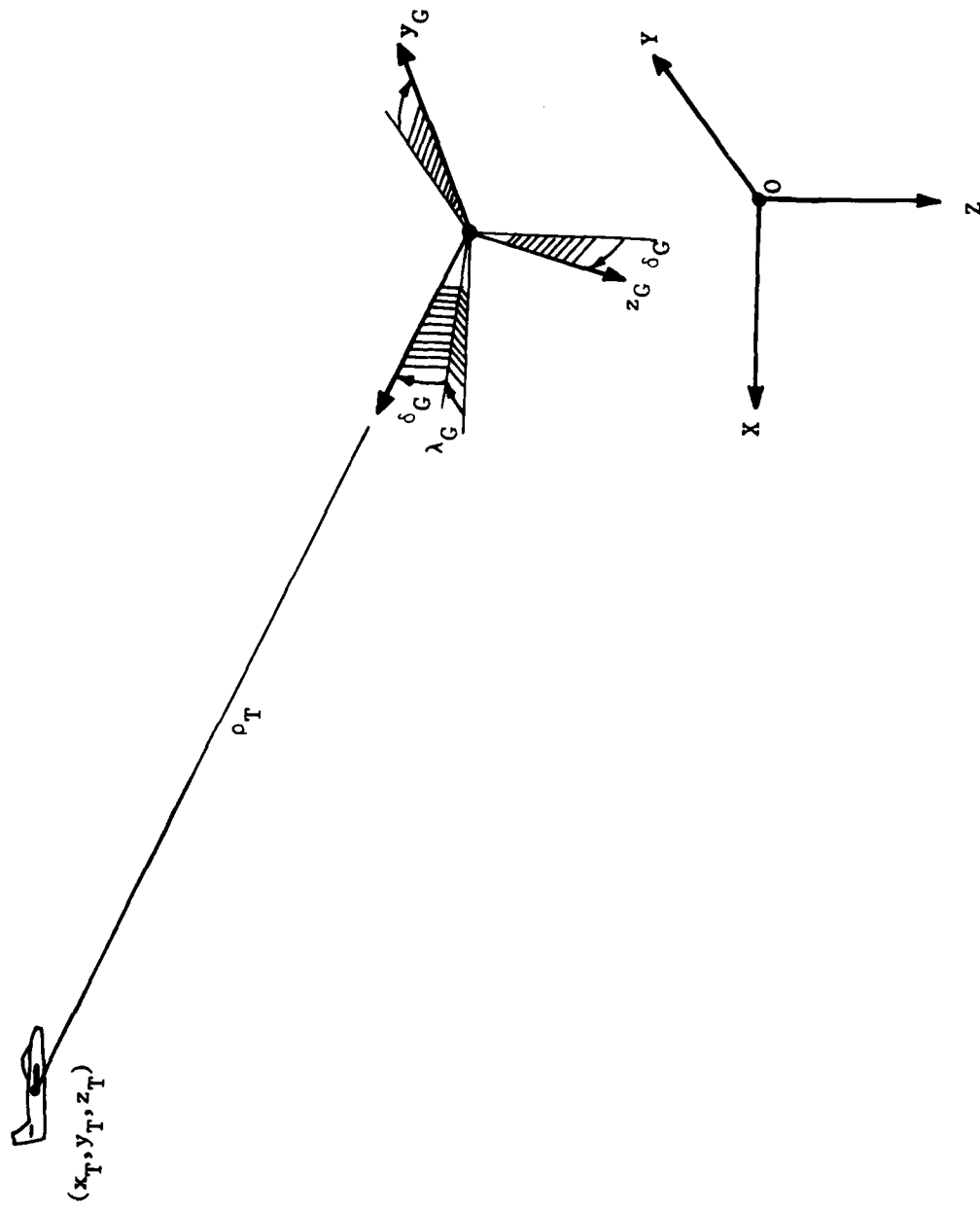


Figure 4-2. Target Location.

respectively, where

$$x_T = x_{Y_0} + \dot{x}_T(t-t_0), \quad (4-5a)$$

$$y_T = y_{T_0} + \dot{y}_T(t-t_0), \quad (4-5b)$$

$$z_T = z_{T_0} + \dot{z}_T(t-t_0), \quad (4-5c)$$

$$\rho_T = \sqrt{x_T^2 + y_T^2 + z_T^2} \quad (4-5d)$$

and x_T , y_T and z_T are the coordinates of the target in a non-rotating coordinate system with origin at G. It follows from Eqs. (4-4) that

$$\dot{\lambda}_G = (\dot{y}_T/x_T - y_T \dot{x}_T/x_T^2) \cos^2 \lambda_G \quad (4-6a)$$

and

$$\dot{\delta}_G = (-\dot{z}_T/\rho_T + z_T \dot{\rho}_T^2/\rho_T^3) / \cos \delta_G \quad (4-6b)$$

where $\dot{\rho}_T = (x_T \dot{x}_T + y_T \dot{y}_T + z_T \dot{z}_T) / \rho_T$.

4.3 Control

Control of the rocket is physically accomplished, as mentioned previously, by causing the deflector vanes to dwell in either a positive or negative position, thereby producing a torque about the control axis (see Fig. 3-4) of the rocket. This torque produces an angular acceleration about either the positive or negative control axis. The integrated effects of the torque are, first of all, an angular rate and, secondly, an attitude change. The latter results in a change of the rocket's angle of incidence and an accompanying change in the aerodynamic force and torque on the rocket. Finally, the altered aerodynamic force produces a change in the rocket's flight path. The duration of the dwell and the roll orientation of the rocket while it is taking place determine the resultant change in its flight path.

The purpose of the control system is two-fold. The first part of its purpose is to "stabilize" the rocket's angular velocity in such a way that angular rates due to disturbances, and not guidance commands, are nulled out rapidly. Secondly, it must produce the attitude changes dictated by the guidance commands.

Since the control force is either "on" or "off", the control system is of the bang-bang type. Thus, in formulating a control algorithm the questions of when to turn the control force on (and in what direction) and how long to leave it on (the dwell time) must be answered.

Control Algorithm

We let $|\Delta t_y|$ and $|\Delta t_z|$ denote the dwell times of the deflector vanes for control of the rocket's motion in the $x_G z_G$ - and $x_G y_G$ -planes, respectively. The signs Δt_y and Δt_z determine whether $F_c = \pm F_{c_{\max}}$, as will be explained in what follows. We also let δ_y and δ_z denote the angles through which the rocket rolls during a "longitudinal" ($x_G z_G$ -plane), or "lateral" ($x_G y_G$ -plane), control period. To be more specific, we assume that positive control torques are desired. Then, to control longitudinal motion, the control force is given by (see Fig. 4-3)

$$F_c = F_{c_{\max}}, \text{ if } 2n\pi - \delta_y/2 \leq \phi + \delta \leq 2n\pi + \delta_y/2, \quad (4-7a)$$

and by

$$F_c = -F_{c_{\max}}, \text{ if } (2n+1)\pi - \delta_y/2 \leq \phi + \delta \leq (2n+1)\pi + \delta_y/2. \quad (4-7b)$$

Here $n = 0, 1, 2, \dots$. Similarly, for lateral control,

$$F_c = F_{c_{\max}}, \text{ if } [(4n+1)/2]\pi - \delta_z/2 \leq \phi + \delta \leq [(4n+1)/2]\pi + \delta_z/2, \quad (4-8a)$$

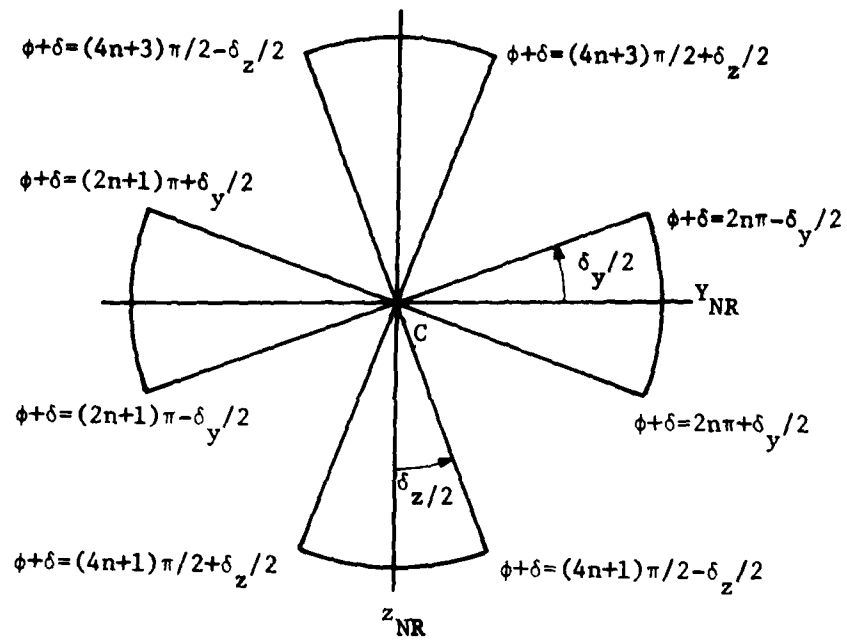


Figure 4-3. Control Regions.

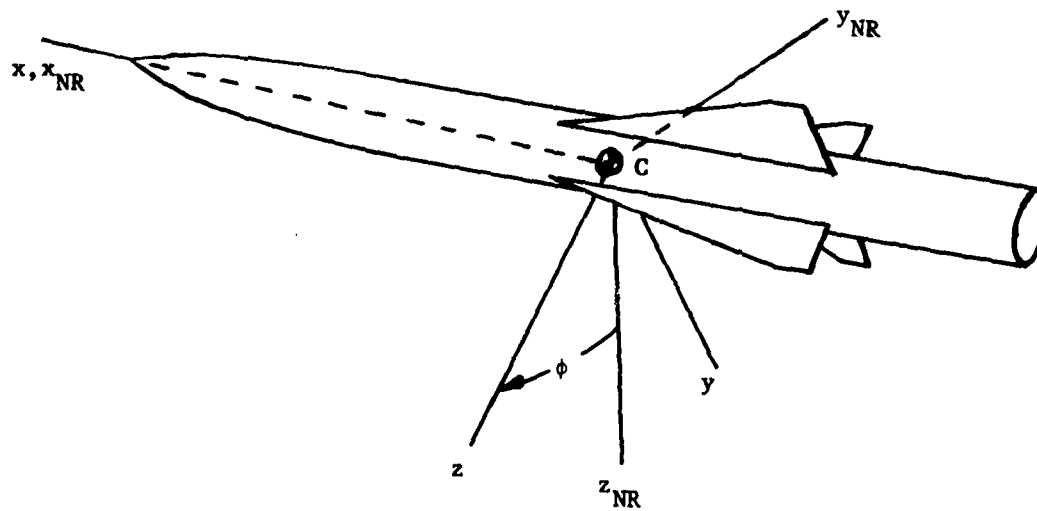


Figure 4-4. Non-rolling Coordinate System.

and

$$F_c = -F_{c_{\max}}, \text{ if } [(4n+3)/2]\pi - \delta_z/2 \leq \phi + \delta \leq [(4n+3)/2]\pi + \delta_z/2. \quad (4-8b)$$

Ideally, the control forces given by Eqs. (4-7) and (4-8) act as space-fixed forces which produce space-fixed torques and the latter result in changes in the rocket's attitude. Since the rocket is rolling, there is some coupling between its pitch and yaw motions. But, since it is also "slender" (i.e., $I_{xx} \ll I_{yy} \approx I_{zz}$), this coupling is not severe.

The roll angle ϕ must be known if F_c is to be generated properly. This angle is assumed to be measured by a roll gyro on board the rocket.

The dwell times Δt_y and Δt_z are determined from the control laws

$$\Delta t_y = -K_1 (Q_{NR} - Q_{NR_c}) \quad (4-9a)$$

and

$$\Delta t_z = -K_1 (R_{NR} - R_{NR_c}), \quad (4-9b)$$

where K_1 is a constant, Q_{NR} is the angular rate of the rocket about the y_{NR} -axis [which is one of the three axes of the non-rolling $Cx_{NR}y_{NR}z_{NR}$ coordinate system (see Fig. 4-4)], R_{NR} is the angular rate of the rocket about the z_{NR} -axis and Q_{NR_c} and R_{NR_c} are commanded angular rates about the y_{NR} - and z_{NR} -axes, respectively. The angular rates Q_{NR} and R_{NR} are computed from

$$Q_{NR} = Q \cos \phi - R \sin \phi \quad (4-10a)$$

and

$$R_{NR} = Q \sin \phi + R \cos \phi, \quad (4-10b)$$

respectively. Also, the commanded angular rates are determined using the equations,

$$Q_{NR_c} = \frac{k_2}{k_1} z_G + \frac{k_3}{k_1} \dot{z}_G \quad (4-11a)$$

and

$$R_{NR_c} = -\frac{k_2}{k_1} y_G - \frac{k_3}{k_1} \dot{y}_G \quad (4-11b)$$

respectively, where K_2 and K_3 are time varying gains which have the forms,

$$K_2 = K_{20} t_o/t \quad (4-12a)$$

and

$$K_{30} = K_{30} t_o/t, \quad (t \geq t_o). \quad (4-12b)$$

Here, K_{20} and K_{30} are constant and t_o is the time the control system is enabled (This is a short time after the rocket exits the tube and corresponds to the time the sustainer motor is ignited.)

We note that, for an actual single-axis system, both Q and R may not be measured by rate gyros. In fact, there will probably be only one rate gyro which measures only the angular rate of the rocket about the control axis. However, δ_y and δ_z are herein restricted to small angles, so that the value of Q_{NR} (R_{NR}) used to calculate Δt_y (Δt_z) is approximately the angular rate about the control axis just prior to a control phase.

In an actual system, values of Q_{NR_c} and R_{NR_c} are calculated by the guidance computer and sent to the rocket via a radio link. These values are used by the rocket's on-board systems to generate Δt_y and Δt_z commands. In the present simulation, the procedure used is: (1) Specify a priori

$\delta_{y_{max}}$ and $\delta_{z_{max}}$ and an associated $\Delta t_{max} = \delta_{y_{max}}/\Omega$, where Ω is the average roll rate of the rocket. (2) Compute Δt_y (Δt_z) and check to see if it is less than Δt_{max} . (3) If $\Delta t_y \leq \Delta t_{max}$ ($\Delta t_z \leq \Delta t_{max}$), then compute $\delta_y = |\Delta t_y| \Omega$ ($\delta_z = |\Delta t_z| \Omega$). (4) If $\Delta t_y > \Delta t_{max}$ ($\Delta t_z > \Delta t_{max}$), then set $\Delta t_y = \text{sign}(\Delta t_y) \Delta t_{max}$ ($\Delta t_z = \text{sign}(\Delta t_z) \Delta t_{max}$) before computing δ_y (δ_z). (5) Check to see if the

rocket's roll attitude satisfies Eqs. (4-7) (Eqs. (4-8)). (6) If it does, control is exercised. If not, no action is taken. (7) If the time since the beginning of this control phase is less than τ , the control force magnitude is adjusted so that a ramp up from zero to $F_{c \max}$ is obtained in τ seconds during the beginning of the phase. Also, if the time before the end of this control phase is less than, or equal to, τ , a ramp down in $F_{c \max}$ is generated during the last τ seconds of the phase. (8) If $|\Delta t_y| < 2\tau$ ($|\Delta t_z| < 2\tau$), no control is exercised.

Control Gain Selection

An approximate linear stability analysis was conducted,⁵ assuming that the rocket's roll rate is rapid enough that the average values of the control torques generated during one period of the rolling motion could be considered continuous torques with respect to the non-rolling system $Cx_{NR} y_{NR} z_{NR}$. This analysis provided some indication of the values of K_1 , K_{20} and K_{30} which should be used, but the responses predicted by the linear analysis were not the same as those obtained via nonlinear simulations. This might be expected, since the control problem is actually very nonlinear. Hence, preliminary values of the gains were chosen using physical and heuristic reasoning and "refined" values obtained by changing the gains selectively and generating trajectories with the simulation code.

The reasoning used to pick the preliminary gains was as follows. The commanded transverse angular rates of the rocket are assumed to be limited by $Q_{c \max}$. Hence, in one control period the control torque total impulse must be no more than that necessary to produce a rate $Q_{c \max}$. For a given maximum control region width, $\delta_{y \max} = \delta_{z \max}$, the duration of control application

is $\Delta t_{\max} = \delta_{y_{\max}} / \Omega$. Hence, if $\delta_{y_{\max}}$ is small, so that the effective magnitude of the control force is approximately $F_{c_{\max}}$, if I_{T_o} is the value of the lesser of I_{yy} or I_{zz} at the beginning of controlled flight and if ℓ_c is the effective moment arm of $F_{c_{\max}}$ about C, then the change in magnitude of the angular rate is approximately (longitudinal control)

$$Q_{c_{\max}} \approx \ell_c F_{c_{\max}} \Delta t_{\max} / I_{T_o} \quad (4-13)$$

Now, from Eq. (4-9a) (with $Q_{NR} = 0$ and $Q_{NR_C} = Q_{c_{\max}}$)

$$\Delta t_{\max} = K_1 Q_{c_{\max}} \quad (4-14)$$

Hence, from Eqs. (4-13) and (4-14), we have

$$K_1 \approx I_{T_o} / (\ell_c F_{c_{\max}}) \quad (4-15)$$

By replacing z_G and \dot{z}_G with their expected maximum values at the beginning of the controlled flight and Q_{NR_C} by $Q_{c_{\max}}$, in Eq. (4-11a), we obtain

$$Q_{c_{\max}} = K_{20} |z_{G_{\max}}| + K_{30} |\dot{z}_{G_{\max}}| \quad (4-16)$$

By considering the results of the linear stability analysis⁵ and some trial simulation results, it was found that equal numerical values for K_{20} and K_{30} resulted in reasonable response characteristics. Hence, (except for units)

$$K_{20} = K_{30} = Q_{c_{\max}} / |z_{G_{\max}}| \quad (4-17)$$

were chosen as preliminary values of the "guidance" control gains.

As stated above, the preliminary values of K_1 , K_{20} and K_{30} were adjusted to obtain values which result in reasonable response characteristics of the typical rocket simulated in this study.

5. SIMULATION RESULTS

5.1 General Comments

To check out the operation of the simulation codes and to study the effects of the various anomalies, such as winds, several simulations were carried out. In the course of these simulations, several different numerical integration algorithms were used to solve the flight phase equations of motion, while the launch phase equations were integrated using only a standard fourth-order Runge-Kutta algorithm. The procedures used to solve the flight phase equations were (1) a variable-step, fourth-order, Runge-Kutta algorithm,⁶ (2) a variable-step Hamming predictor-corrector algorithm,⁶ (3) a fixed-step, fourth-order, Runge-Kutta algorithm and (4) a fixed-step Euler algorithm.

The variable-step algorithms were not satisfactory because, during time intervals in which the control force changes very rapidly, the step-size adjusting segments of the algorithms failed to find a suitable step-size. This occurred, not because of inaccuracies in the calculations, but as a result of the methods used to check on the accuracy of the computations.

The fixed-step algorithms were tested by using various step sizes and a "best" step size was obtained for each. These "best" step sizes are defined as those which produced consistent solutions and which are such that step sizes larger or smaller produced inconsistent results.

Simulations using the fixed-step, Runge-Kutta algorithm with its "best" step size require more computer time than their Euler algorithm counterparts. Hence, the fixed-step, Euler algorithm was used to generate the

results presented herein. However, the simulation code contains both algorithms and the one used is optional with the user.

5.2 Data Used in the Simulations

Most of the data for the launch phase simulations is given in Table 5-1. The remaining data is the angular rate of the launcher turret and the tipoff distance. These were different for each of the cases for which results are given and may be found in Table 5-2.

The data for the flight phase simulations is given in Table 5-3 and Appendix C. The physical characteristics of the rocket are given in Table 5-3. The manner in which aerodynamic parameters were calculated is explained in Appendix C. Values used for the control gains were $K_1 = 0.125 \text{ sec}^2$, $K_{20} = 0.03 \text{ rad-sec/m}$ and $K_{30} = 0.03 \text{ rad-sec}^2/\text{m}$.

5.3 Simulation Results

Since the question of whether the rocket is acquired by the IR sensor and guided in such a manner that radar acquisition is possible is the one of interest, only the flight phase results of the simulations are presented here. The results of the corresponding launch phase simulation provided the initial conditions for each flight phase simulation. The results for ten simulations are given here. These may be classified as follows:

- Simulation 1 - No Anomalies
- Simulation 2 - Tipoff
- Simulation 3 - Mass Imbalance
- Simulation 4 - Thrust Misalignment
- Simulation 5 - Cross Wind
- Simulations 6 and 7 - "Positive" Turret Rates
- Simulations 8 and 9 - "Negative" Turret Rates
- Simulation 10 - "Combination" of Anomalies

TABLE 5-1. LAUNCH PHASE DATA

<u>Rocket</u>	
LENGTH	2.4 (m)
DIAMETER	0.16 (m)
MASS	62.4 (kg)

$$I_x = 0.325 \text{ (kg-m}^2\text{)}$$

$$I_y = I_z = 21.3 \text{ (kg-m}^2\text{)}$$

$$\text{THRUST} \quad 16850.0 \text{ Nt)}$$

$$\text{SPIN TORQUE} \quad 100.0 \text{ (Nt-m)}$$

Other Data

$$\frac{R}{T}A = (-0.95 \quad 1.03 \quad 0.0)^T \text{ (m)}$$

$$\frac{R}{T}G = (-0.95 \quad 0.0 \quad -1.015)^T \text{ (m)}$$

$$\frac{R}{T}C/A(\text{INITIAL}) = (1.001 \quad 0.0 \quad 0.0)^T \text{ (m)}$$

$$\ell_E = 2.4 \text{ (m)}$$

TABLE 5-2. DESCRIPTION OF FLIGHT CONDITIONS.

RUN NUMBER	TIPOFF DISTANCE (METERS)	TURRET ANGULAR RATE (DEG/SEC)	MASS IMBALANCE ($\times 10^{-3}$)		THRUST MISALIGNMENT ($\times 10^{-3}$)			WIND VELOCITY IN INERTIAL SYSTEM (METERS/SEC)		
			μ_2	μ_3	α_y	α_z	α_z	u_w	v_w	w_w
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.7	0.7	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.35	0.35	0.35	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-6.0	0.0
6	0.0	+5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	+10.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	-5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	-10.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.15	+10.0	0.7	0.7	0.35	0.35	0.35	0.0	-6.0	0.0

TABLE 5-3. FLIGHT PHASE DATA.

REFERENCE AREA FOR AERODYNAMICS - $0.021 \text{ (m}^2\text{)}$

REFERENCE LENGTH FOR AERODYNAMICS - 0.16 (m)

TIME OF BOOSTER BURNOUT - 1.65 (sec)

MASS - (INITIAL) - 63.2 (kg)

MASS - (AT BOOSTER BURNOUT) - 58.3 (kg)

MOMENTS OF INERTIA (INITIAL) - $I_x = 0.325 \text{ (kg-m}^2\text{)}$

$$I_y = I_z = 21.5 \text{ (kg-m}^2\text{)}$$

MOMENTS OF INERTIA (AT BOOSTER BURNOUT) - $I_x = 0.299 \text{ (kg-m}^2\text{)}$

$$I_y = I_z = 15.46 \text{ (kg-m}^2\text{)}$$

\underline{u} (INITIAL) - $(1.001 \ 0.0 \ 0.0)^T \text{ (m)}$

\underline{u} (AT BOOSTER BURNOUT) - $(1.110 \ 0.0 \ 0.0)^T \text{ (m)}$

\underline{e}_T - $(0.124 \ 0.0 \ 0.0)^T \text{ (m)}$

$|x'_1|$ - 0.158 (m)

x'_2 - 0.793 (m)

SLAT - 0.0 (m)

I_{sp} - 230.0 (sec)

$F_{c_{max}}$ - 444.822 (Nt)

δ - 0.0 (rad)

THRUST VALUES - $16850.0, 16850.0, 18471.5, 18471.5, 1621.5, 1621.5,$
 0.0 (Nt)

THRUST TIMES - $0.0, 0.200, 0.201, 1.650, 1.651, 13.350, 13.351 \text{ (sec)}$

No Anomalies

In Simulation 1 the turret is non-rotating, there is no tipoff, no wind and the rocket is "perfect," i.e., no mass imbalance nor thrust misalignment. The results for this simulation are shown in Figs. 5-1 through 5-17. The first four figures, Figs. 5-1 through 5-8, show Δt_y , Q_{NR_c} , Q_{NR} , Q , Δt_z , R_{NR_c} , R_{NR} and R as functions of time. These figures illustrate the guidance and control process. As the rocket leaves the launch tube, the force due to gravity causes it to drop and hence a pitch up command is needed (see Fig. 5-2). However, when the rocket leaves the launch tube, guidance is not possible because the rocket's aft end is not within the FOV of the IR sensor. From the time of sustainer motor ignition to the time the rocket's aft end is within the FOV of the IR sensor, Q_{NR_c} and R_{NR_c} are zero and the rocket is only rate stabilized. Since its transverse angular rates at the beginning of the flight phase were zero for this perfect launch situation, no action was taken by the control system.

For this simulation, guidance commands were possible after about 0.40 seconds. Due to the rocket's orientation at that time, lateral control was first exerted, then longitudinal control. Note that Δt_y and Δt_z had maximum magnitude at the start of guided flight, but not for the total simulation. Note that Q_{NR_c} (Fig. 5-2) is composed of "pulses" for which the maximum values vary smoothly with time. Note also that the amplitude of the oscillatory curve obtained by connecting the midpoints of the extremities of adjoining steps has the appearance of a damped oscillation.

The time history of the angular rate about the y_{NR} axis, Q_{NR} , is shown in Fig. 5-3. Note the rapid response in Q_{NR} . Also, of interest are the

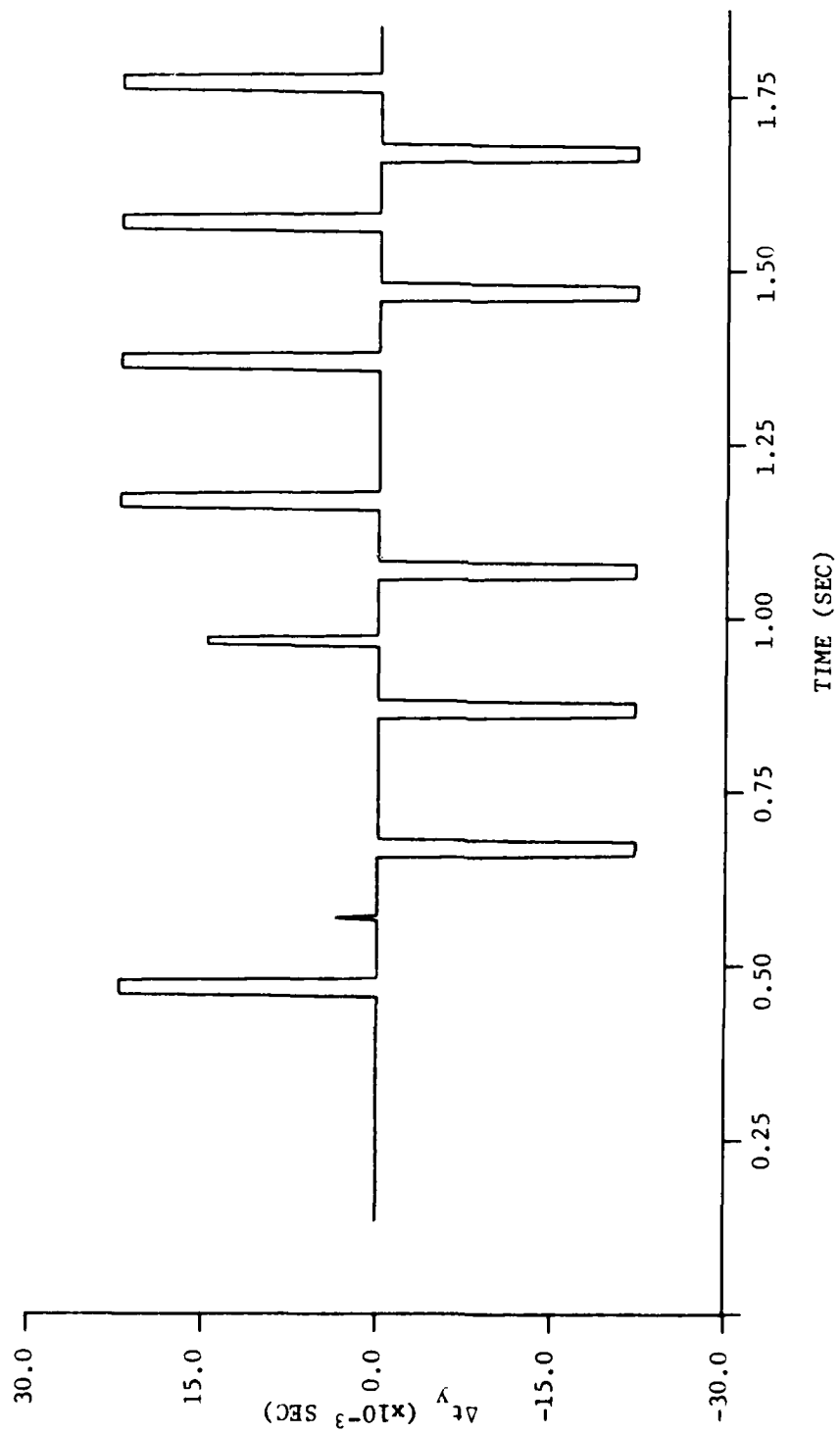


Figure 5-1. Pitch Control Time, Δt_y , for Simulation 1.

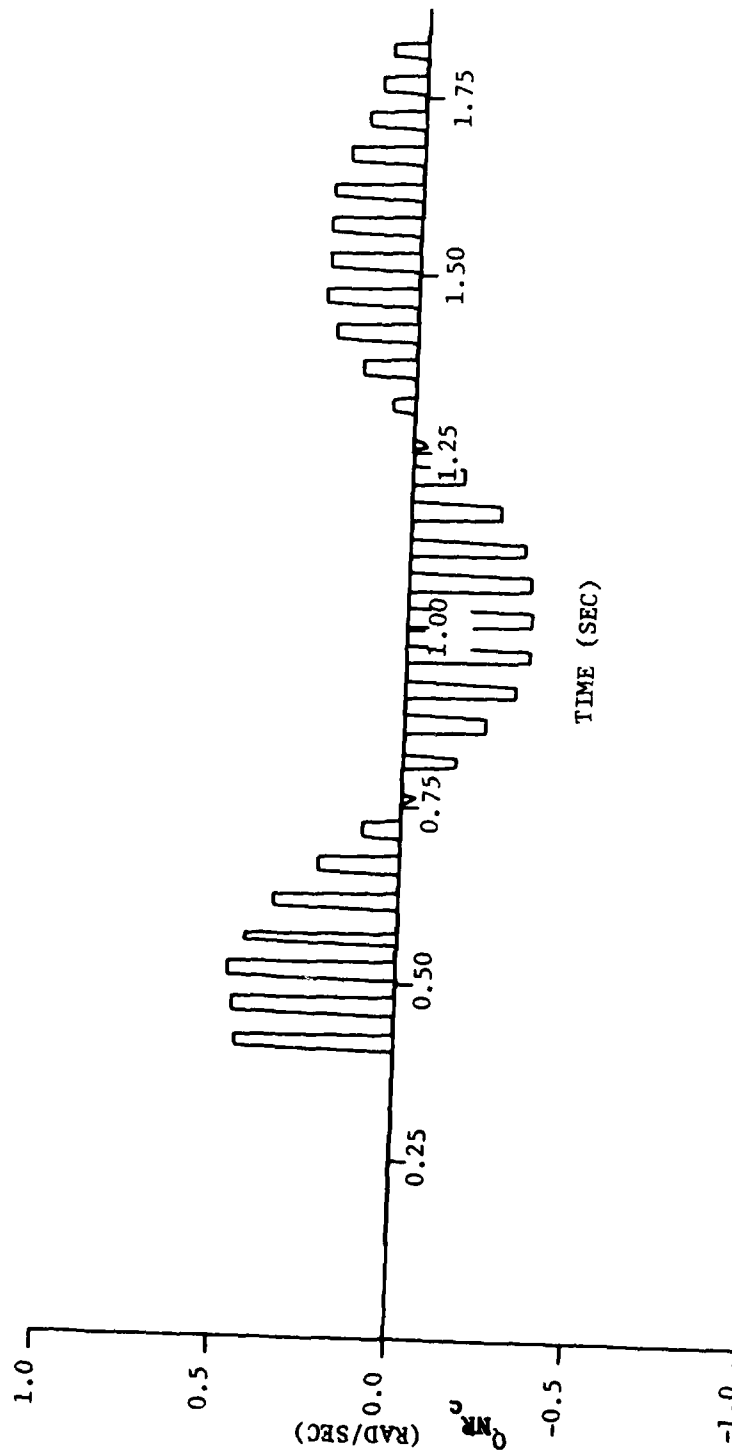


Figure 5-2. Commanded Non-rolling Angular Rate, Q_{NR_c} , for Simulation 1.

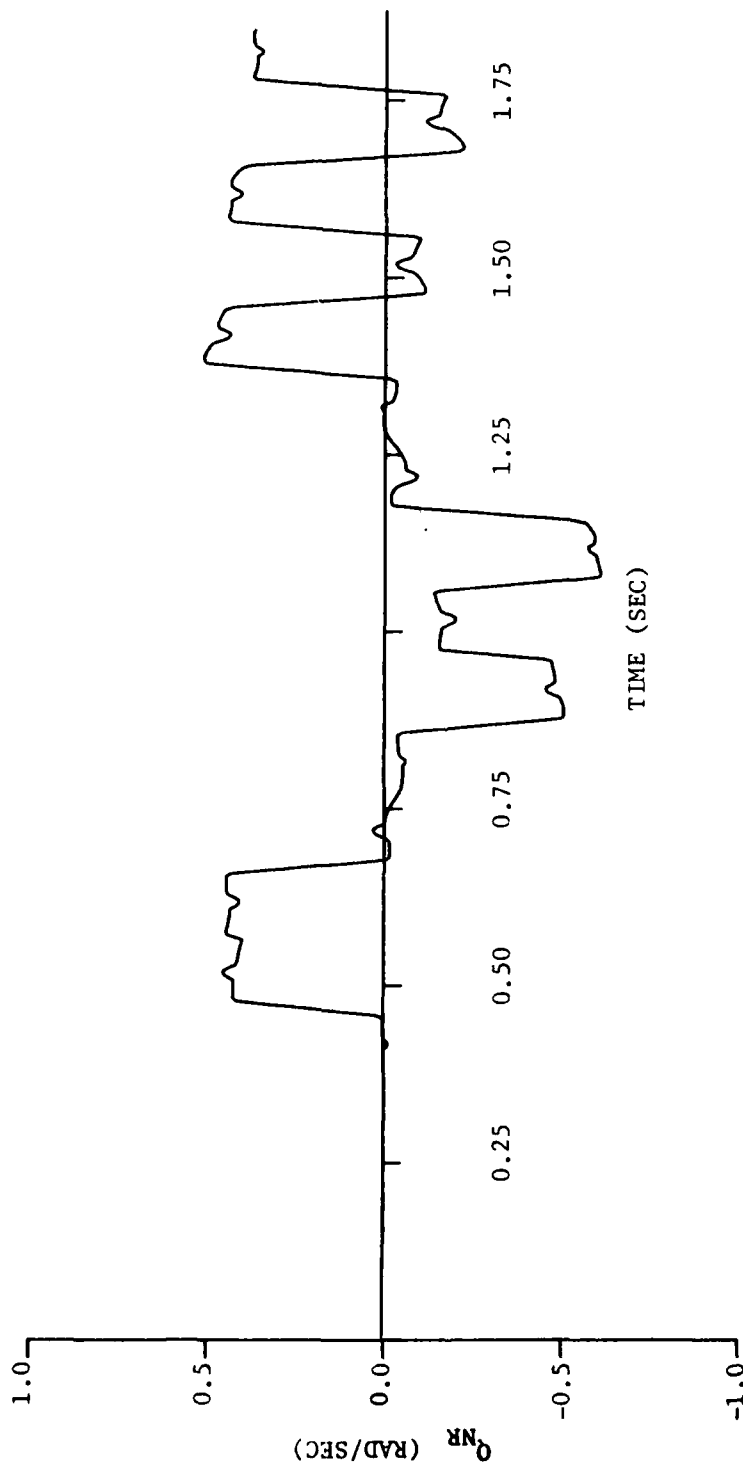


Figure 5-3. Non-rolling Angular Rate, ϕ_{NR} , for Simulation 1.

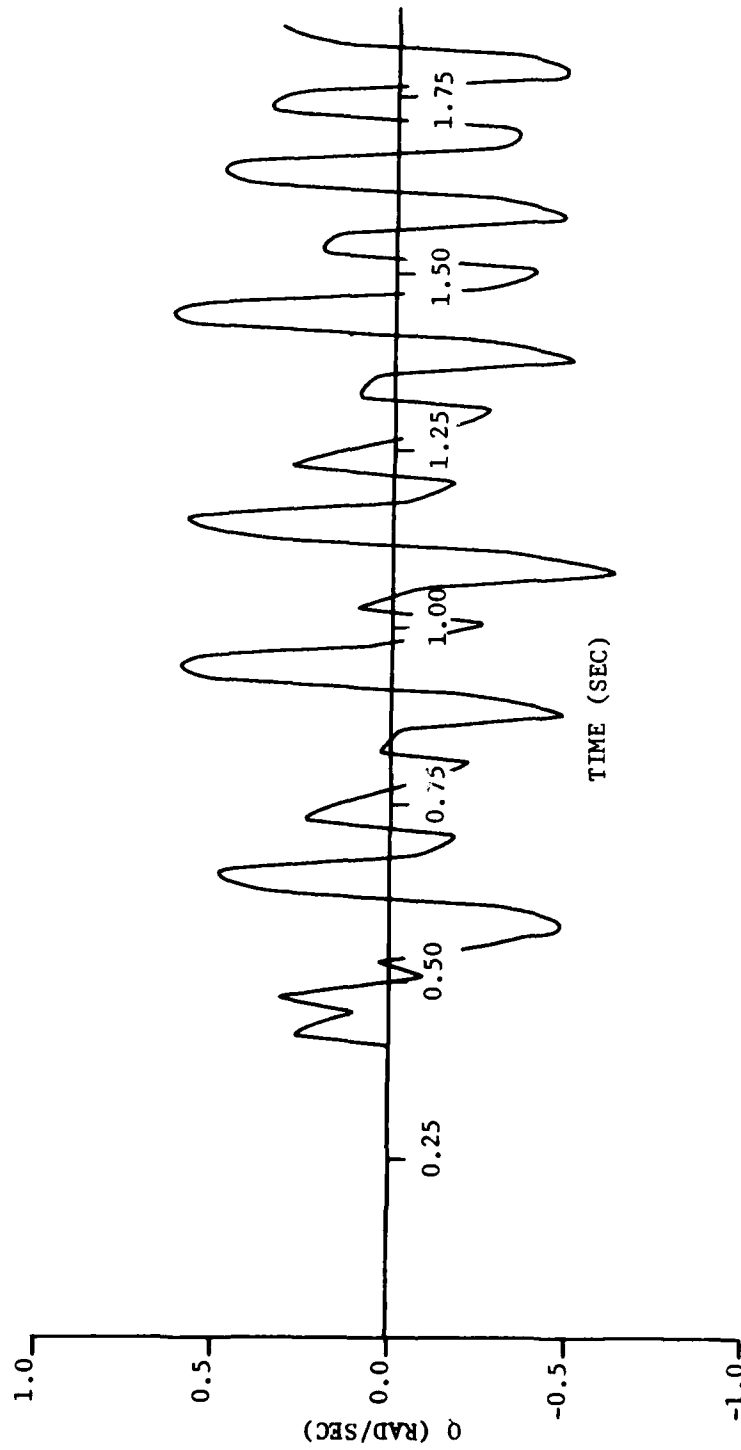


Figure 5-4. Rocket-Fixed Angular Rate, Q , for Simulation 1.

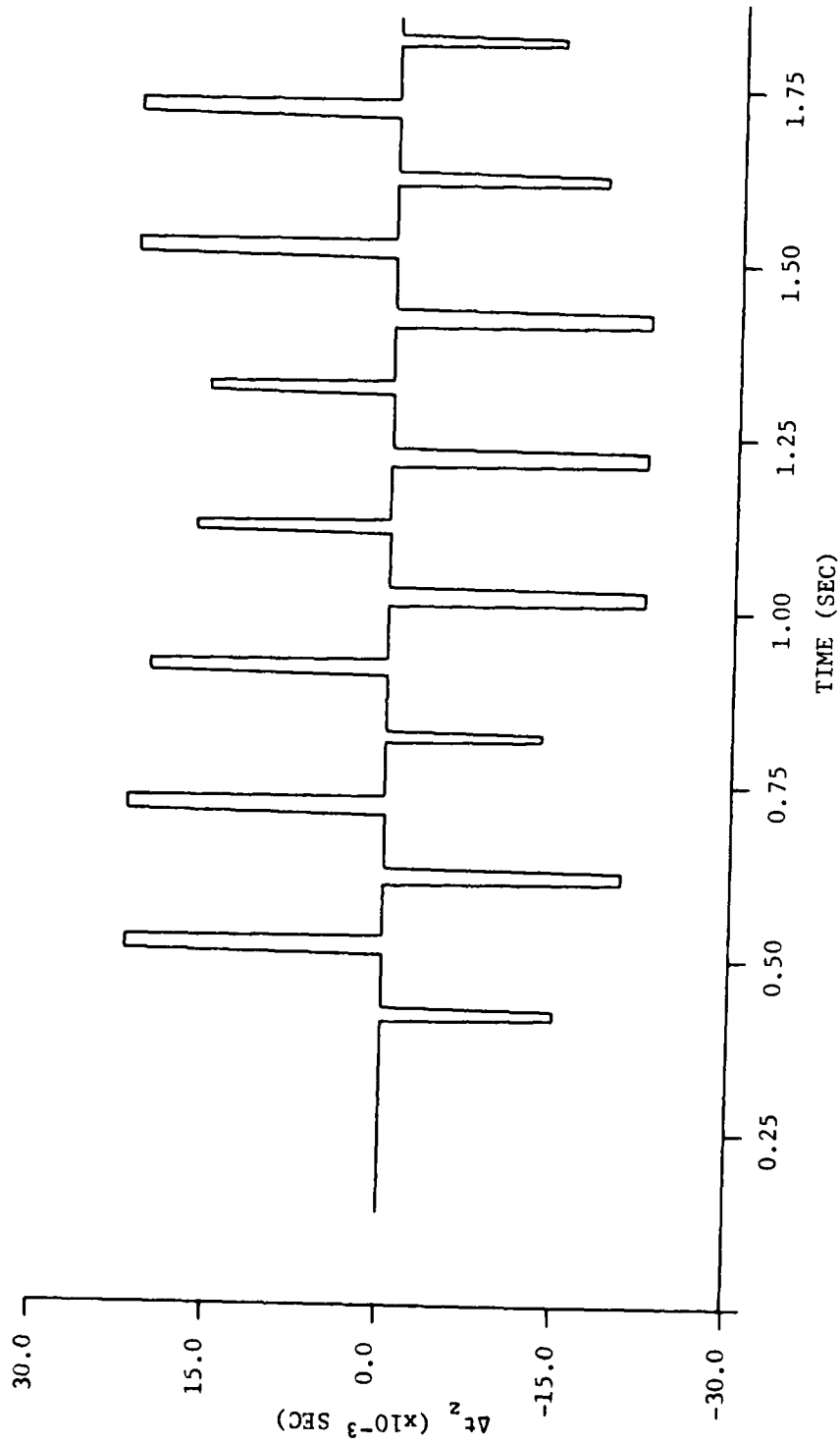


Figure 5-5. Yaw Control Time, Δt_y , for Simulation 1.



Figure 5-6. Commanded Non-rolling Angular Rate, R_{NR_c} , for Simulation 1.

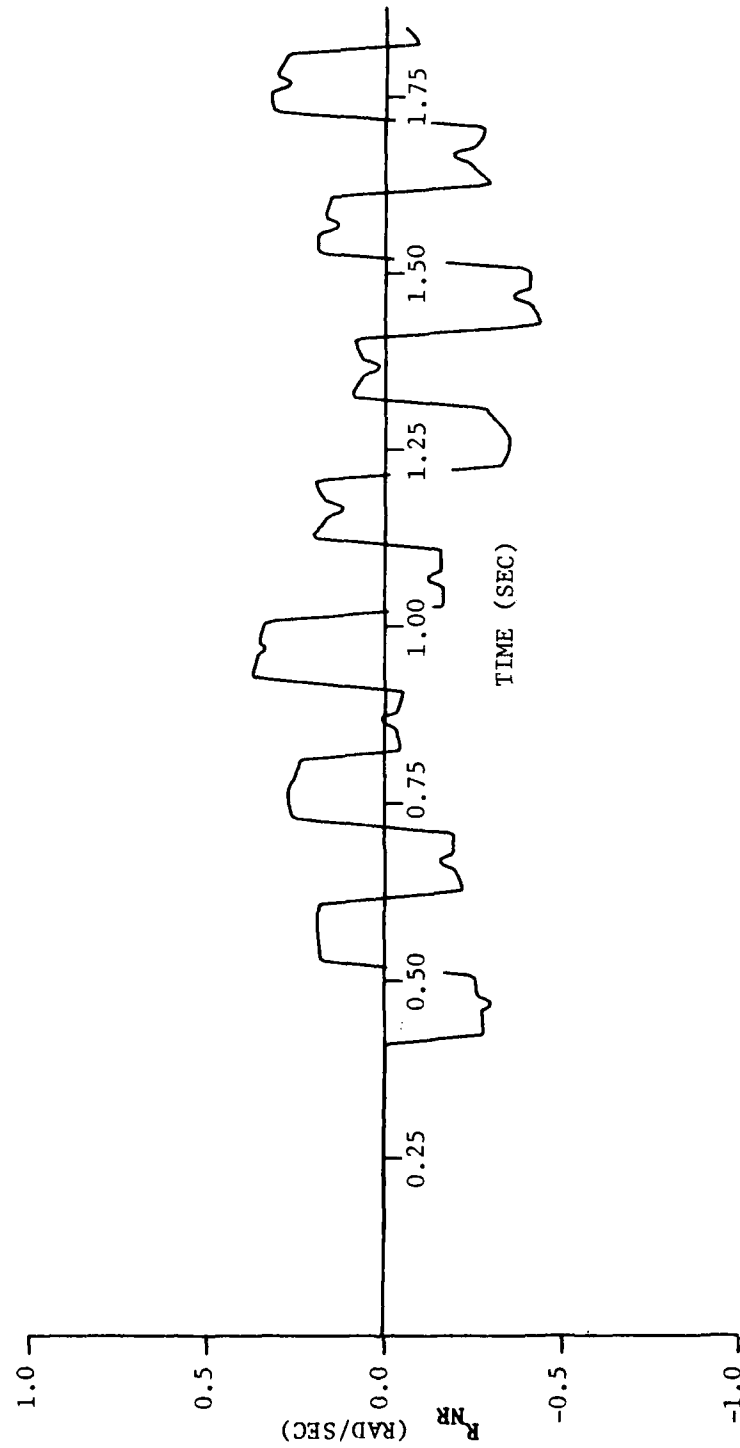


Figure 5-7. Non-rolling Angular Rate, R_{NR} , for Simulation 1.

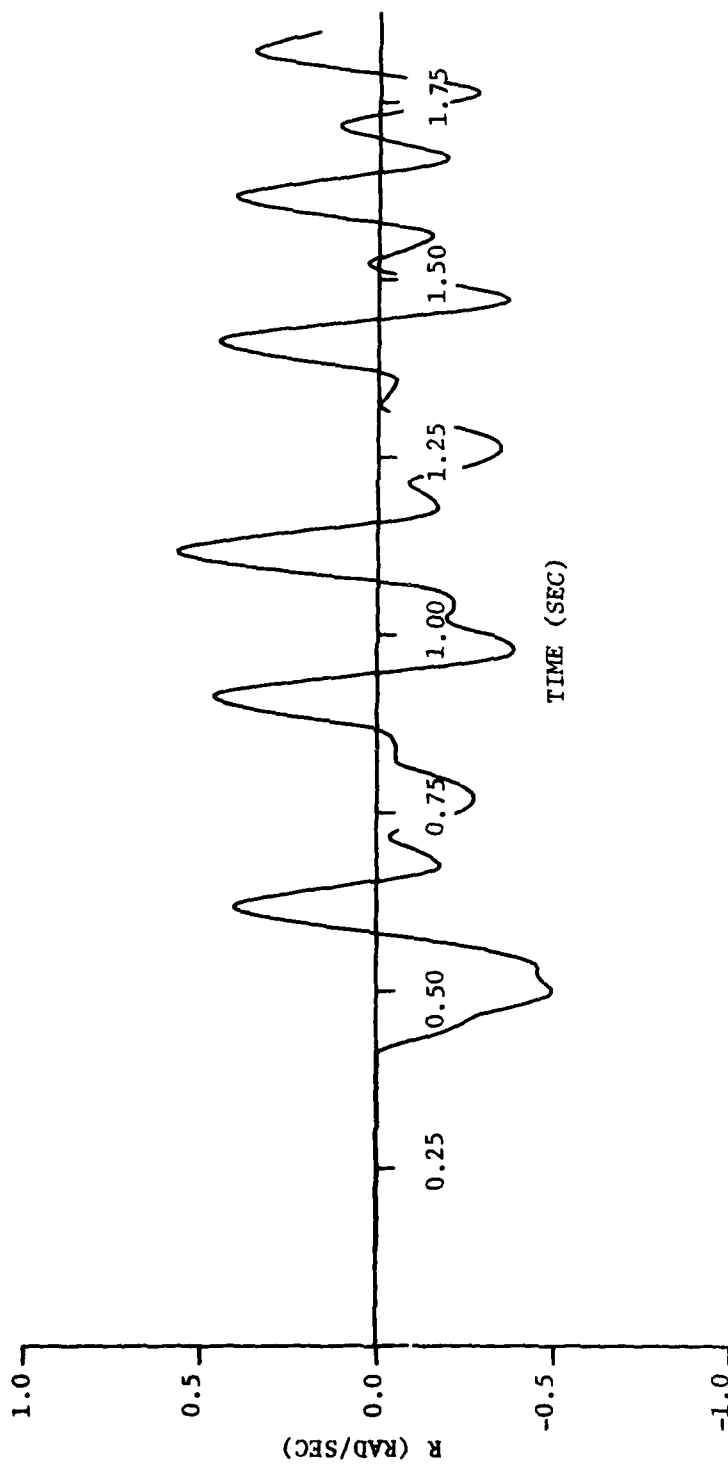


Figure 5-8. Rocket-Fixed Angular Rate, R , for Simulation 1.

small "bumps" and the Q_{NR} curve. These are due to the application of control forces during lateral control phases, since a finite control region centered on $\phi + \delta = [(2n+1)/2]\pi$, for example, with Δt_z positive, results in first a small negative torque about that axis and then a positive torque. The net effect of each pair of positive and negative torques is essentially zero. We note that if there were no aerodynamic damping of transverse angular rates, the tops of the "pulses" in Q_{NR} would (except for the bumps) be flat between control force applications.

The angular rate about the y-axis is shown in Figure 5-4. Because $\delta=0$ for these simulations, the control axis is the y-axis. Therefore, rapid changes in Q with time occur when control of either a longitudinal or lateral motion of the rocket takes place.

The results for Δt_z , R_{NR_c} and R_{NR} are similar to those for Δt_y , Q_{NR_c} and Q_{NR} , respectively. However, the rates R_{NR_c} and R_{NR} are generally smaller than their "longitudinal" counterparts. The angular rate about the z-axis, R , is a much smoother function of time than Q , since it results from the integrated effect of control torques applied about the y-axis.

Figures 5-9 and 5-10 are plots of the angles α_x and β_x as functions of time. Since the wind was zero for this simulation, $\alpha_x = \alpha$ and $\beta_x = \beta$ and hence, these are the time histories of the angle of attack and side slip angle, respectively, of the rocket. Note that α and β are 90° out of phase, as should be the case.

The pitch and yaw angles of the rocket as functions of time are shown in Figs. 5-11 and 5-12. In this simulation the "target" was stationary and could be considered to be located on a line of sight with an elevation of

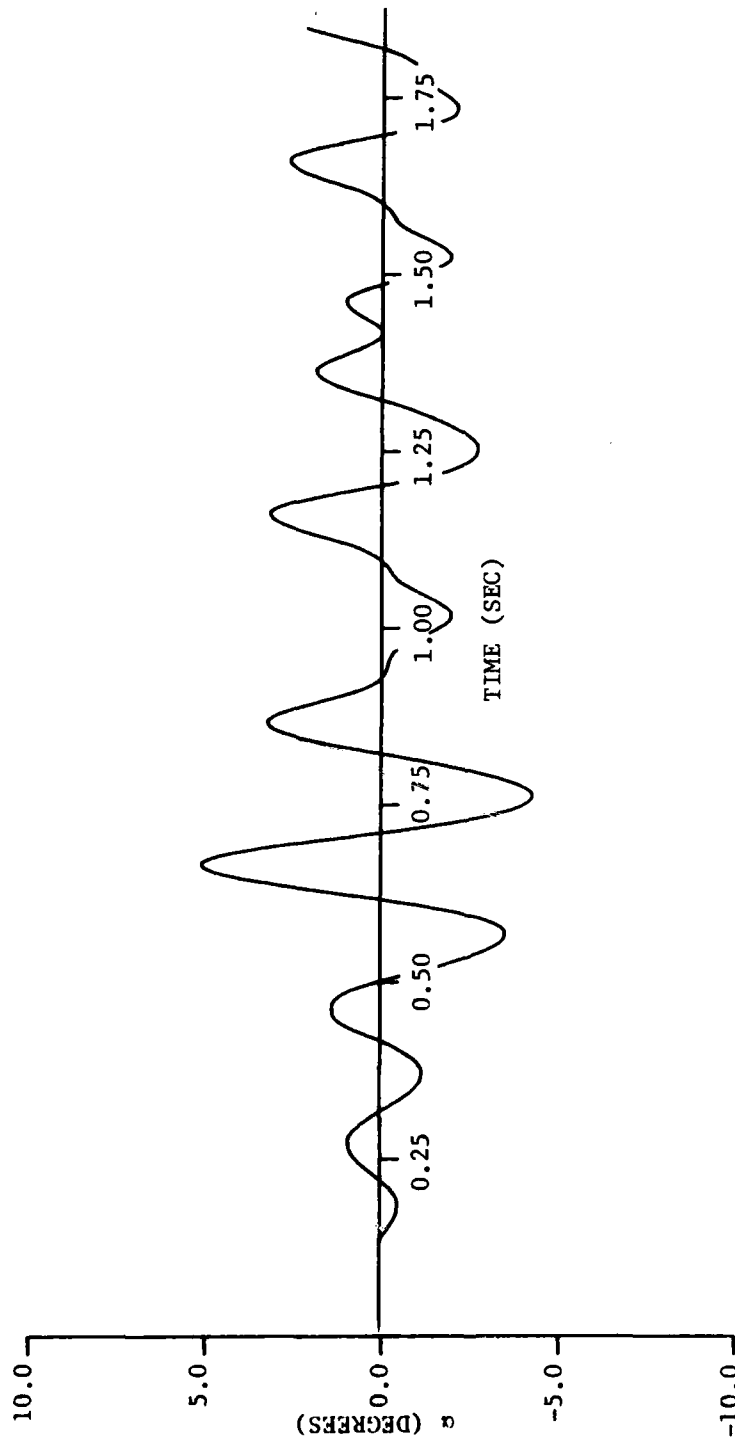


Figure 5-9. Angle of Attack Time History.

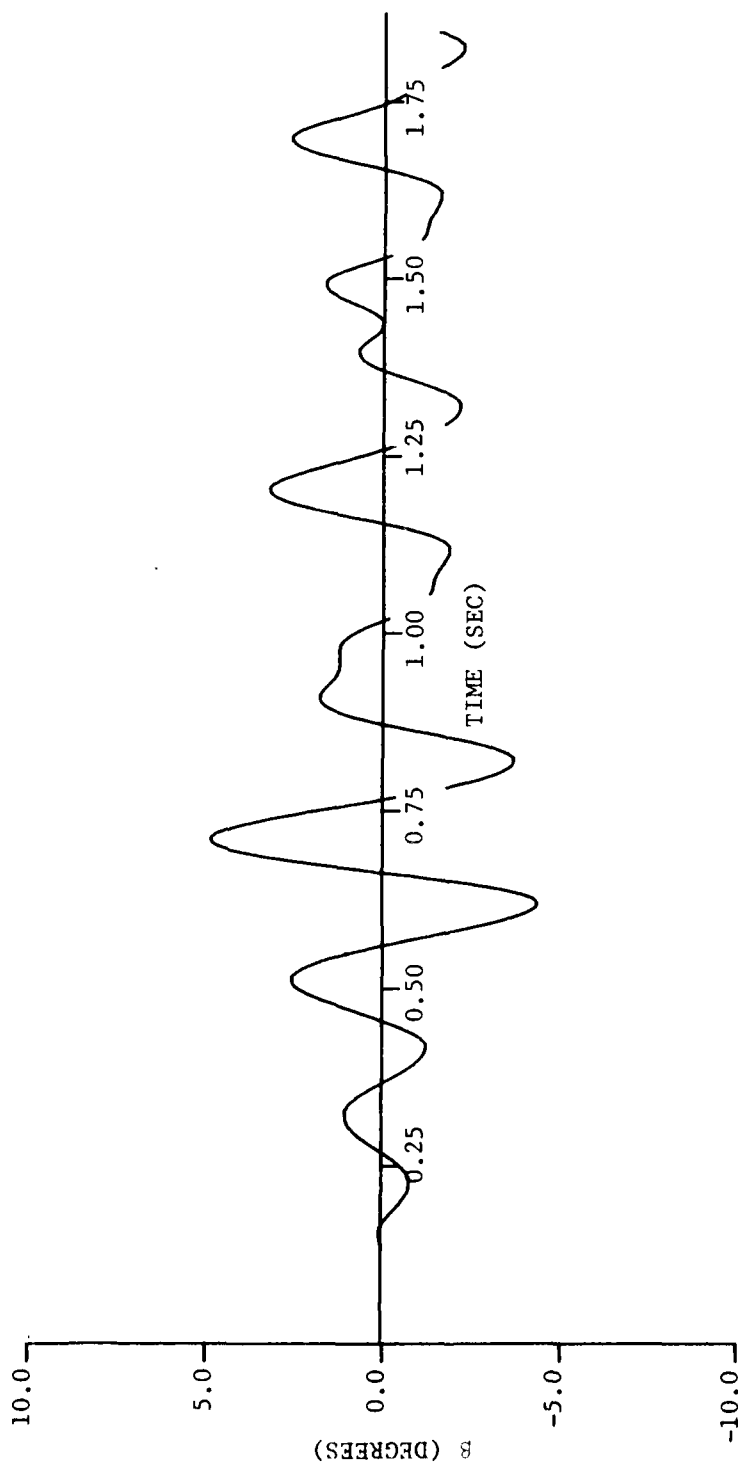


Figure 5-10. Sideslip Angle Time History.

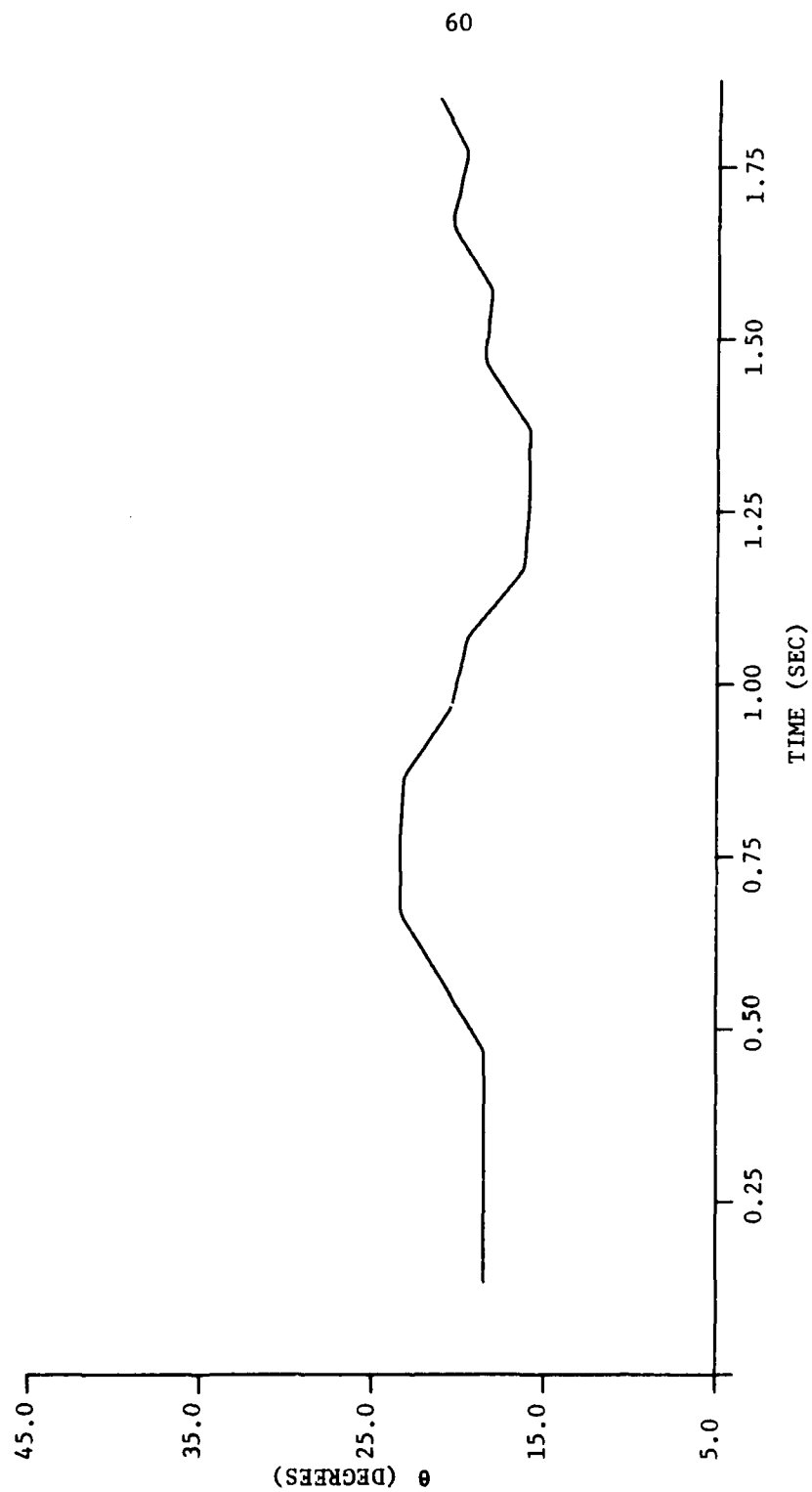


Figure 5-11. Pitch Angle Time History.

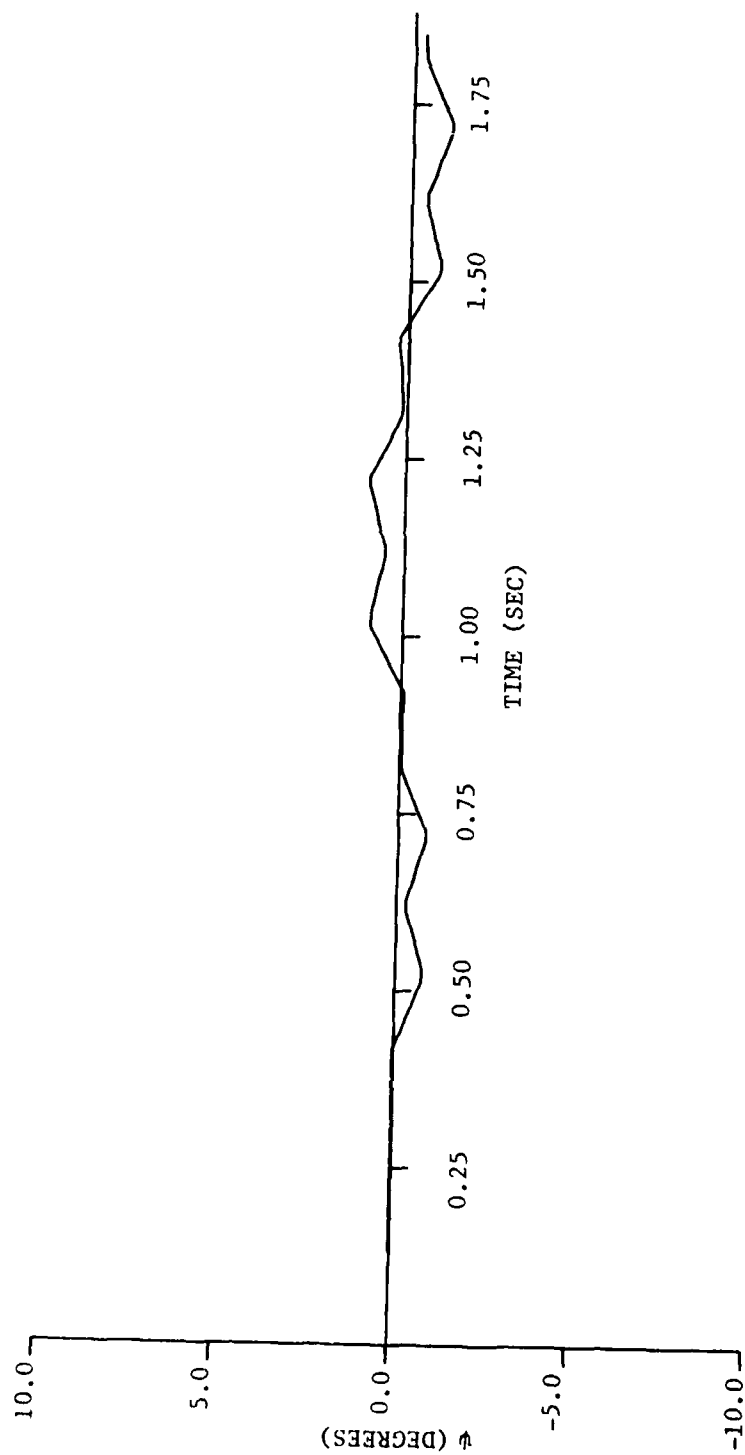


Figure 5-12. Yaw Angle Time History.

18.5°. The rocket's pitch angle was therefore 18.5°, initially. Due to the "gravity drop" as the rocket left the launch tube with zero aerodynamic incidence angle, the guidance and control systems operated to produce an increase in θ for a period of time during which the rocket reacquired, and overshoot the line of sight. Long period oscillations in θ , with smaller short period oscillations superimposed then occurred. The yaw angle ψ was always small with mean value approximately zero. Figure 5-13 is included because it shows clearly the maximum and minimum values of θ and ψ which occurred.

The x_G -coordinate of the center of mass of the rocket as a function of time is presented in Fig. 5-14. It is essentially a cubic function of time during the portion of flight simulated.

Three "views" of the trajectory of the center of mass of the rocket as seen in the $Gx_G y_G z_G$ system are given in Fig. 5-15. Also shown in Fig. 5-15 is the boundary of the 1° FOV of the IR sensor. This is the smaller of two FOV's used, the larger being 4.5°. We note that although the transverse angles are not very smooth functions of time, the trajectory of the rocket's center of mass is very smooth. The "head-on" view given in Fig. 5-15 shows that the center of mass of the rocket stays within 2 meters of the launcher to target LOS during the course of the simulation.

Tipoff

For the simulation in which tipoff from the launcher was included and for Simulations 3-10, only the three views of the trajectory are given. The three views of the trajectory for the tipoff simulation are shown in Fig. 5-16.

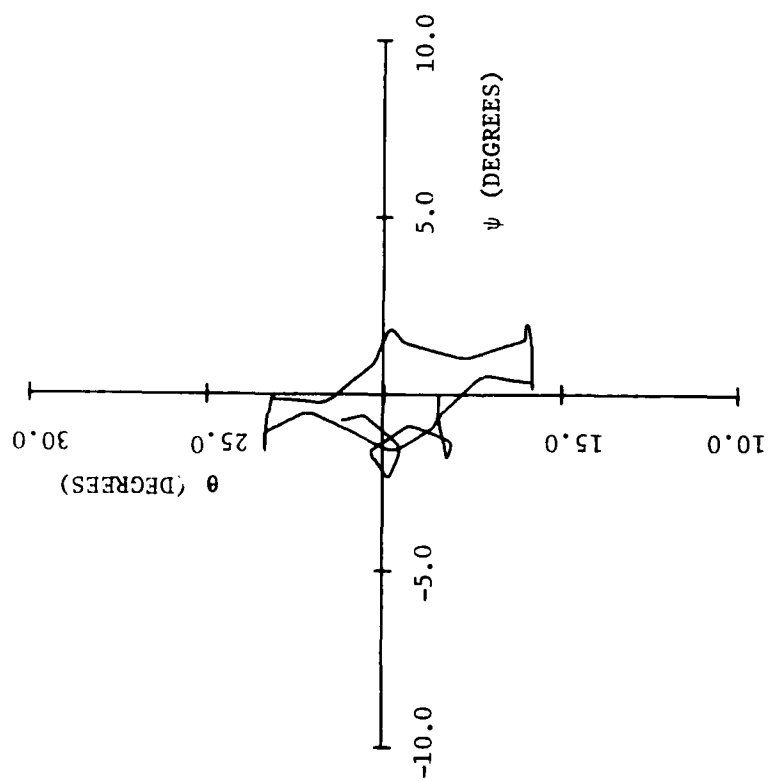


Figure 5-13. Yaw Angle vs. Pitch Angle.

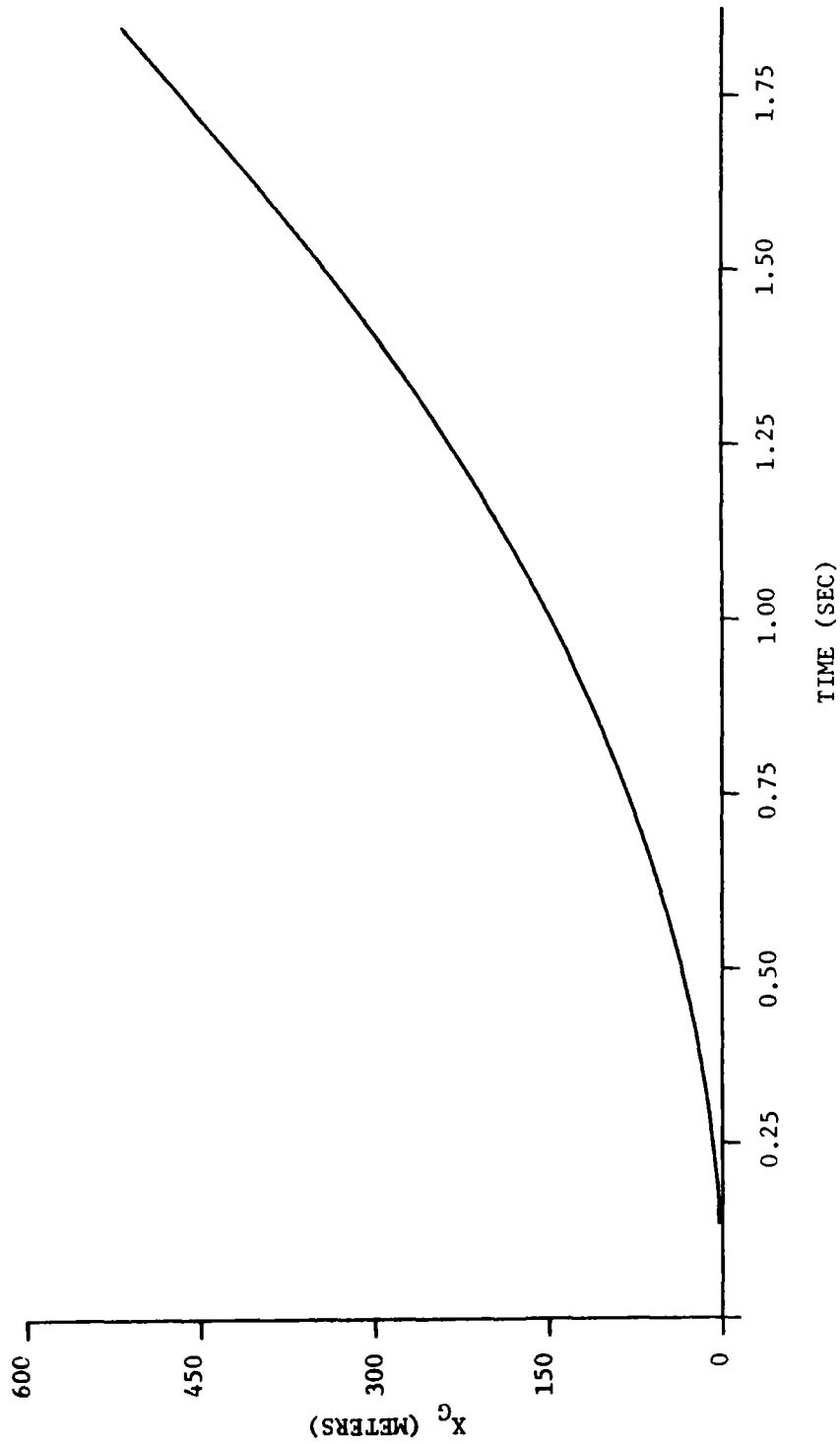


Figure 5-14. Time History of x_G -Coordinate of C.

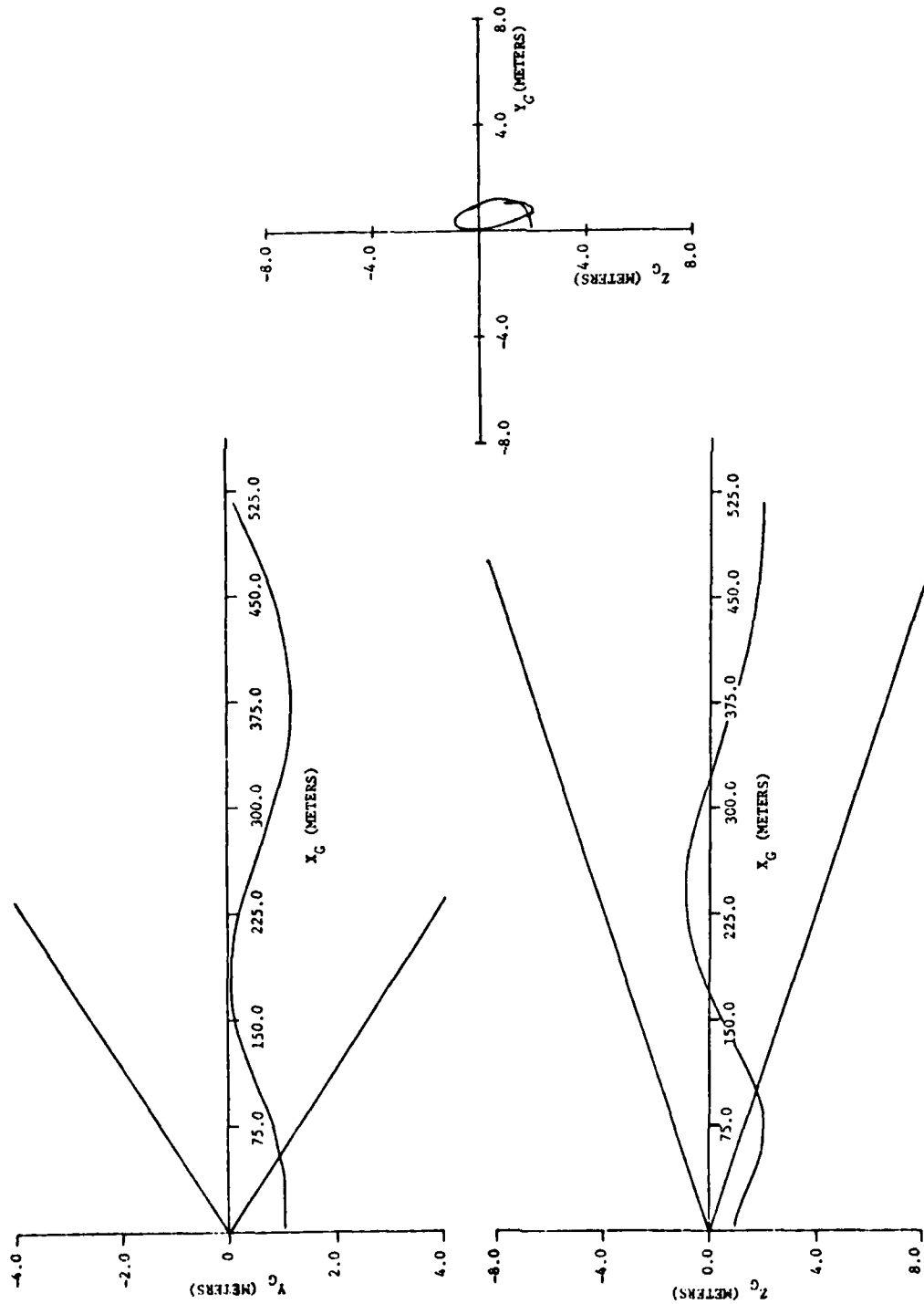


Figure 5-15. Trajectory for Simulation 1.

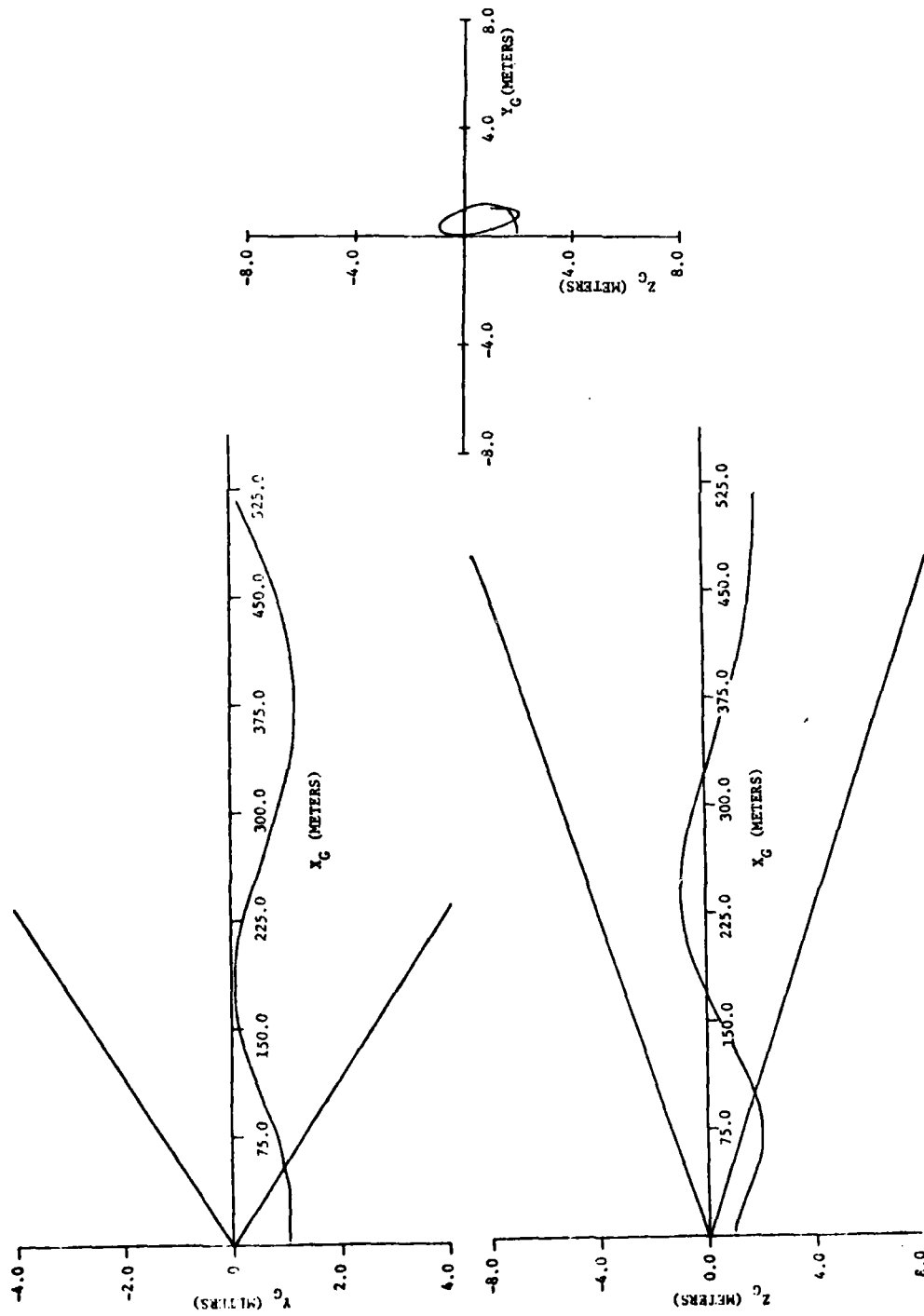


Figure 5-16. Trajectory for Simulation 2.

The main effects of tipoff are, of course, that the rocket has in initial transverse angular rate and its center of mass is also moving transversely when it clears the launch tube. Upon comparing the trajectory for the nominal simulation (Simulation 1) with that given in Fig. 5-16, we note little difference in the two sets of figures. Hence, by itself, tipoff (at least for a distance less than or equal to 0.15 m) is not significant.

Mass Imbalance

Since no rocket is perfect, each may be expected to be dynamically unbalanced. Simulation 3 was made to determine whether such mass imbalance would have a significant effect on the rocket's trajectory. In the simulation code the angles μ_2 and μ_3 (which are small angles of rotation about the y- and z-axes, respectively) were used to generate time varying products of inertia,

$$I_{xy} = -\mu_3(I_{yy} - I_{xx}) \quad (5-1a)$$

and

$$I_{xz} = \mu_2(I_{zz} - I_{xx}). \quad (5-1b)$$

For $\mu_2 = \mu_3 = .0007$, the three views of the trajectory of the rocket's center of mass shown in Fig. 5-17 were produced via simulation. This particular source of error above does not appear to present any problem as far as guidance and control of the rocket are concerned.

Thrust Misalignment

Figure 5-18 shows three views of the trajectory generated by including only thrust misalignment as an error source. For the values, $\alpha_y = \alpha_z = .00035$,

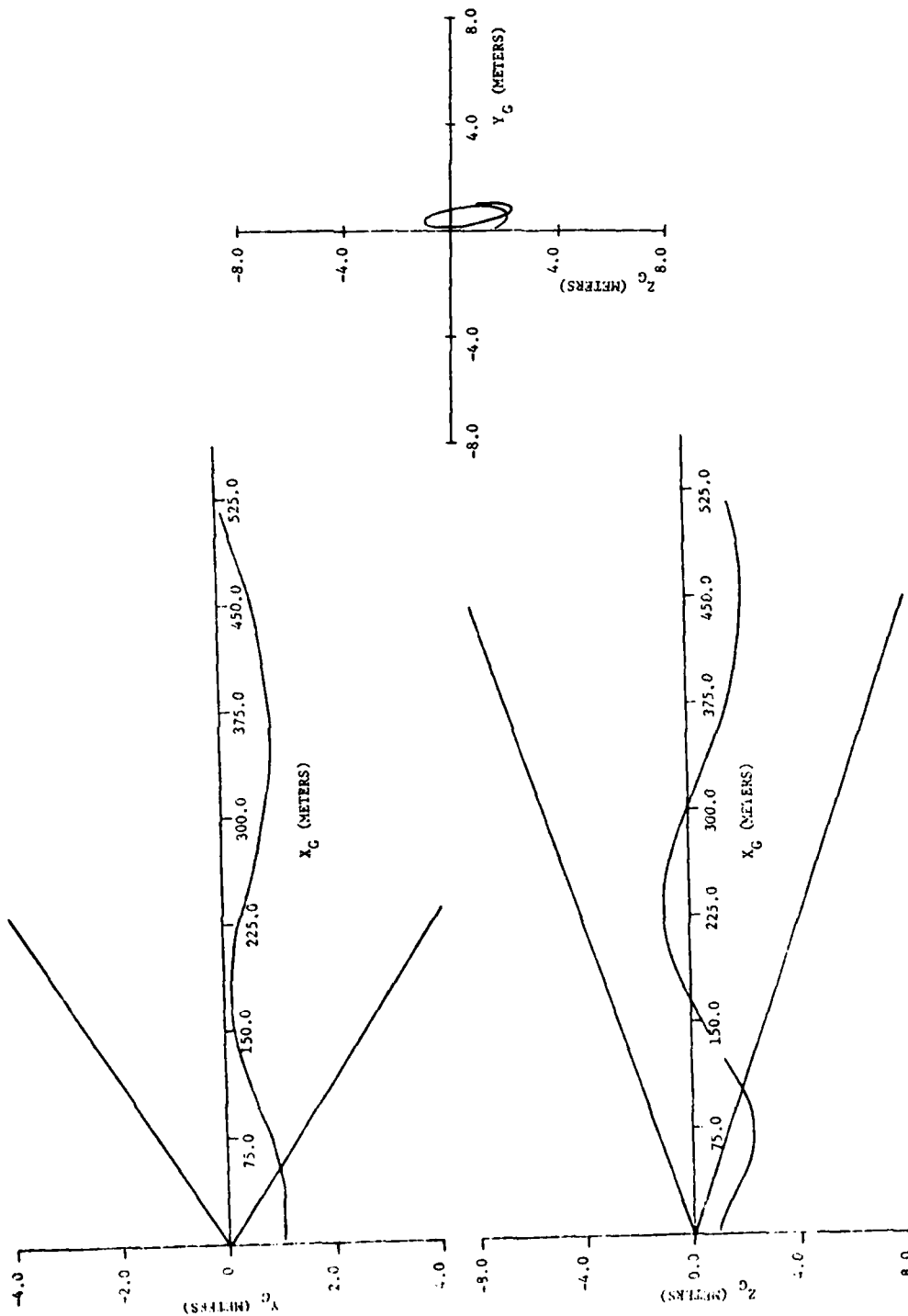


Figure 5-17. Trajectory for Simulation 3.

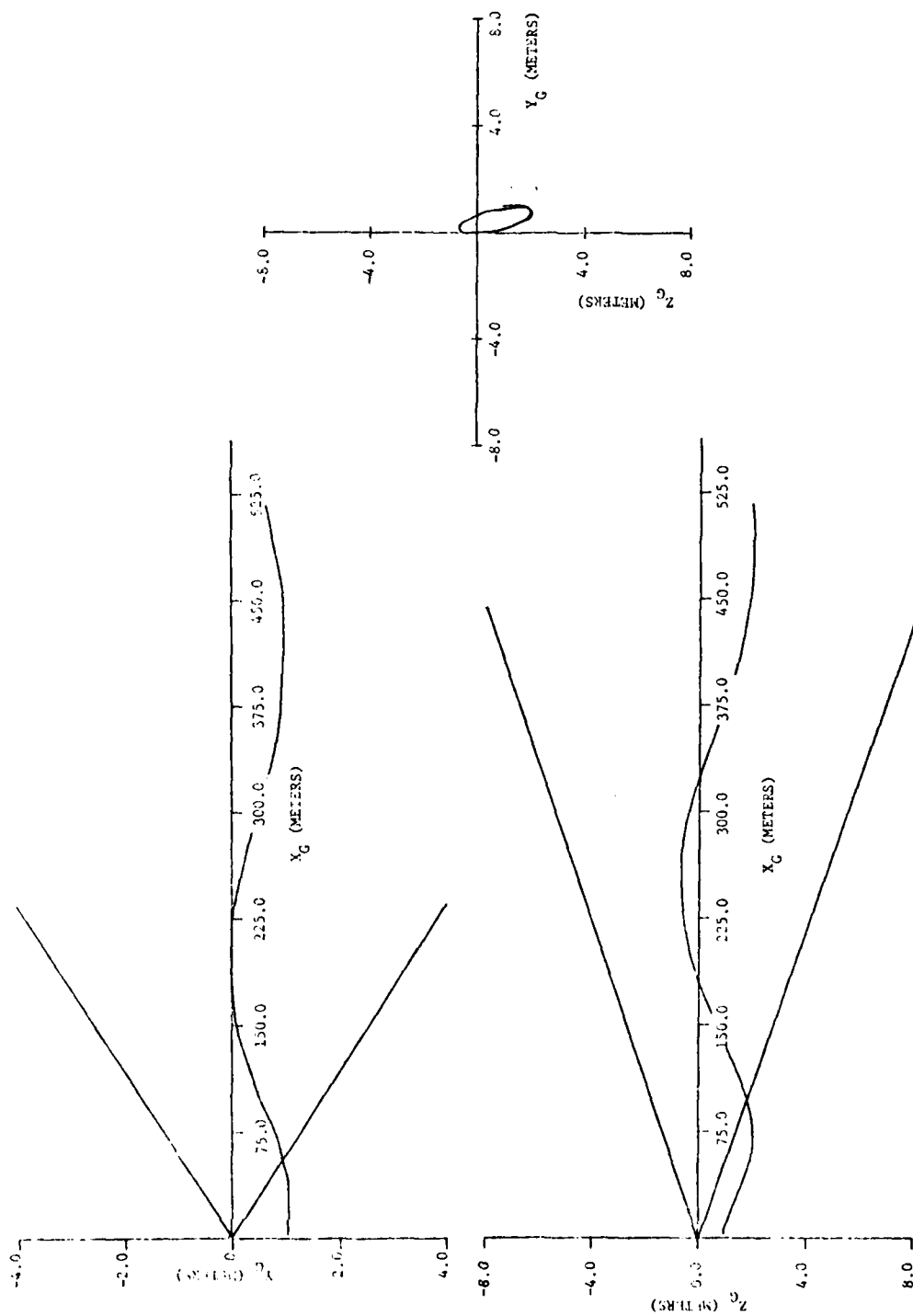


Figure 5-18. Trajectory for Simulation 4.

we note that thrust misalignment is also not a severe problem. That is, although there are recognizable differences in the trajectories for Simulations 1 and 4, these are not large enough to cause concern.

Cross Wind

Several simulations were conducted to determine the relative effects of cross winds on the motion of the rocket. Simulation 5 is representative of the results obtained. In 5, a steady cross wind of 6 meters/second in the -Y-direction was included in the input for an otherwise nominal run. By comparing the trajectory shown in Fig. 5-19, with the nominal trajectory (Fig. 5-15) small differences may be noted. In particular, the rocket (since it is statically stable) turns into the wind. This is seen clearly by comparing Figs. 5-15 and 5-19.

Turret Angular Rates

To track a moving target, the turret of the launch must rotate about the vertical axis z_T . The rapidity of this rotation depends on the speed of the target, its distance from the launcher and its direction of flight. The $Gx_G y_G z_G$ system also rotates about its y_G -axis during the tracking process. In this study, we have neglected this rate of rotation in the launch phase simulations, but not in the flight phase simulations.

For Simulations 6 and 7, the target's motions were such that the initial (for flight phase) angular rates of the launcher turret about the z_T -axis were $+5^\circ/\text{sec}$ and $+10^\circ/\text{sec}$, respectively. The target's initial position for the two cases was (1800, 0, -600) km and the velocities were $[302 \ 157 \ 0]^T$ m/sec and $[130 \ 314 \ 0]^T$ m/sec, respectively. The rocket was fired, in both cases, from the right-hand tube of the launcher. Thus, in these simulations

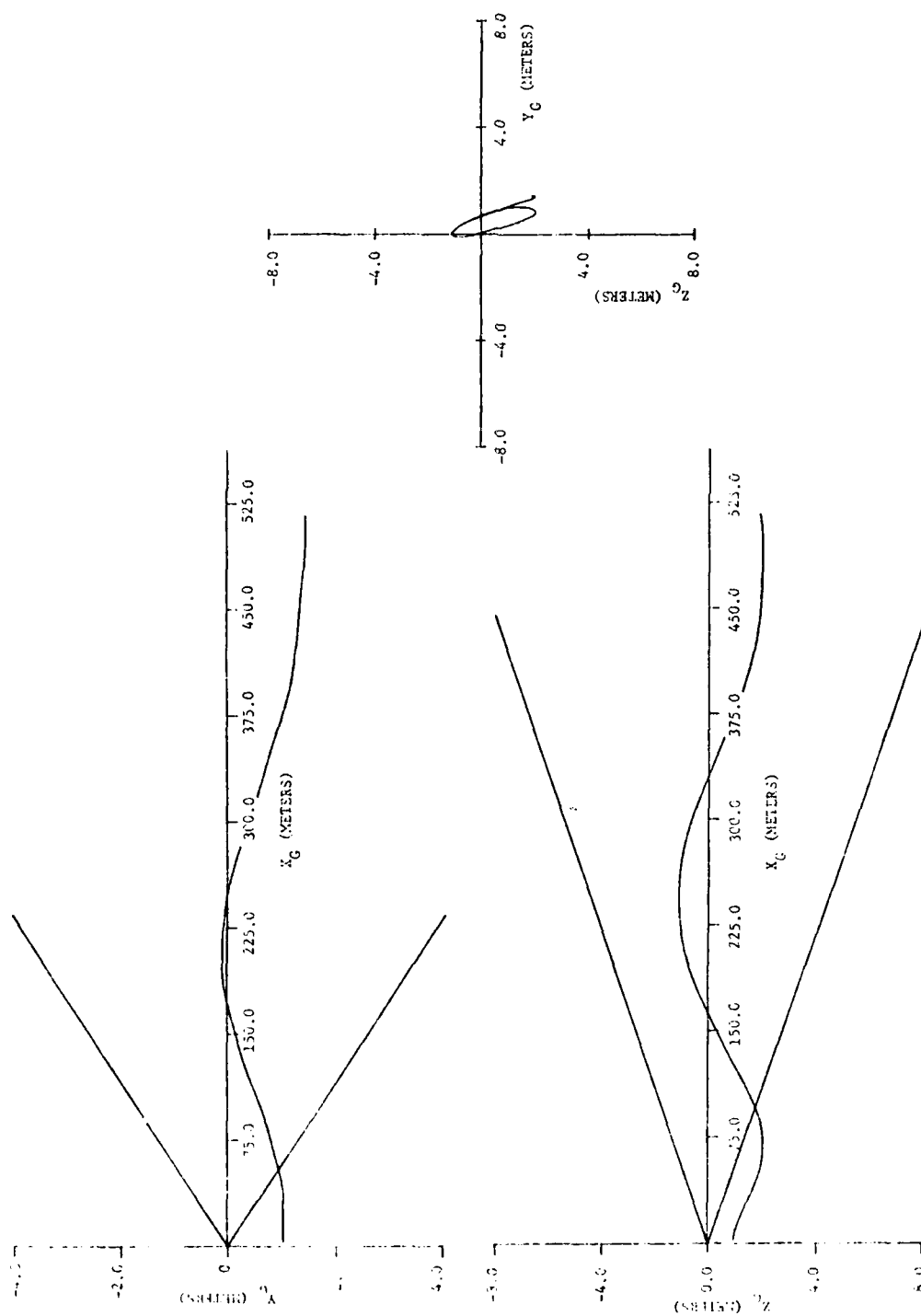


Figure 5-19. Trajectory for Simulation 5.

the FOV of the IR sensor was rotated so that it tended to "engulf" the rocket. This resulted in the entry of the rocket into the 4.5° FOV more quickly than in the nominal case (Simulation 1). Hence, guidance was begun sooner after launch in Simulations 6 and 7 than in the nominal case. The trajectories for Simulations 6 and 7 are shown in Fig. 5-20 and Fig. 5-21, respectively. These results indicate that the system simulated would perform adequately even for the $10^\circ/\text{sec}$ turret angular rate if the rocket were fired from the right-hand, or "leading" tube.

Results for negative turret angular rates about the z_T -axis are shown in Fig. 5-22 and Fig. 5-23, for cases in which the initial rates were $-5^\circ/\text{sec}$ and $-10^\circ/\text{sec}$, respectively. These were obtained by using the data used in Simulations 6 and 7, except the target velocities. These velocities were replaced by $[301 \ -157 \ 0]^T$ and $[130 \ -314 \ 0]^T$ m/sec, respectively, for Simulations 8 and 9. The rocket was acquired by the IR sensor when the initial rate was approximately $-5^\circ/\text{sec}$, since its aft end did come within the wider, 4.5° FOV of the sensor (see Fig. 5-22). Also, the guidance and control system quickly brought the rocket within the narrower, 1° FOV of the sensor and it remained therein. However, in the $-10^\circ/\text{sec}$ case, the IR sensor's FOV was rotated away from the rocket's initial direction of flight so rapidly that, although the rocket entered the 4.5° FOV momentarily, it did not remain therein. Hence, guidance and control of the rocket was lost.

Combination of Anomalies

Simulation 10 was made with a combination of anomalies as indicated in Table 5-1. As can be seen in Fig. 5-24, the rocket entered the 4.5° FOV and remained within the 1° FOV after reaching a range of approximately 310 m.

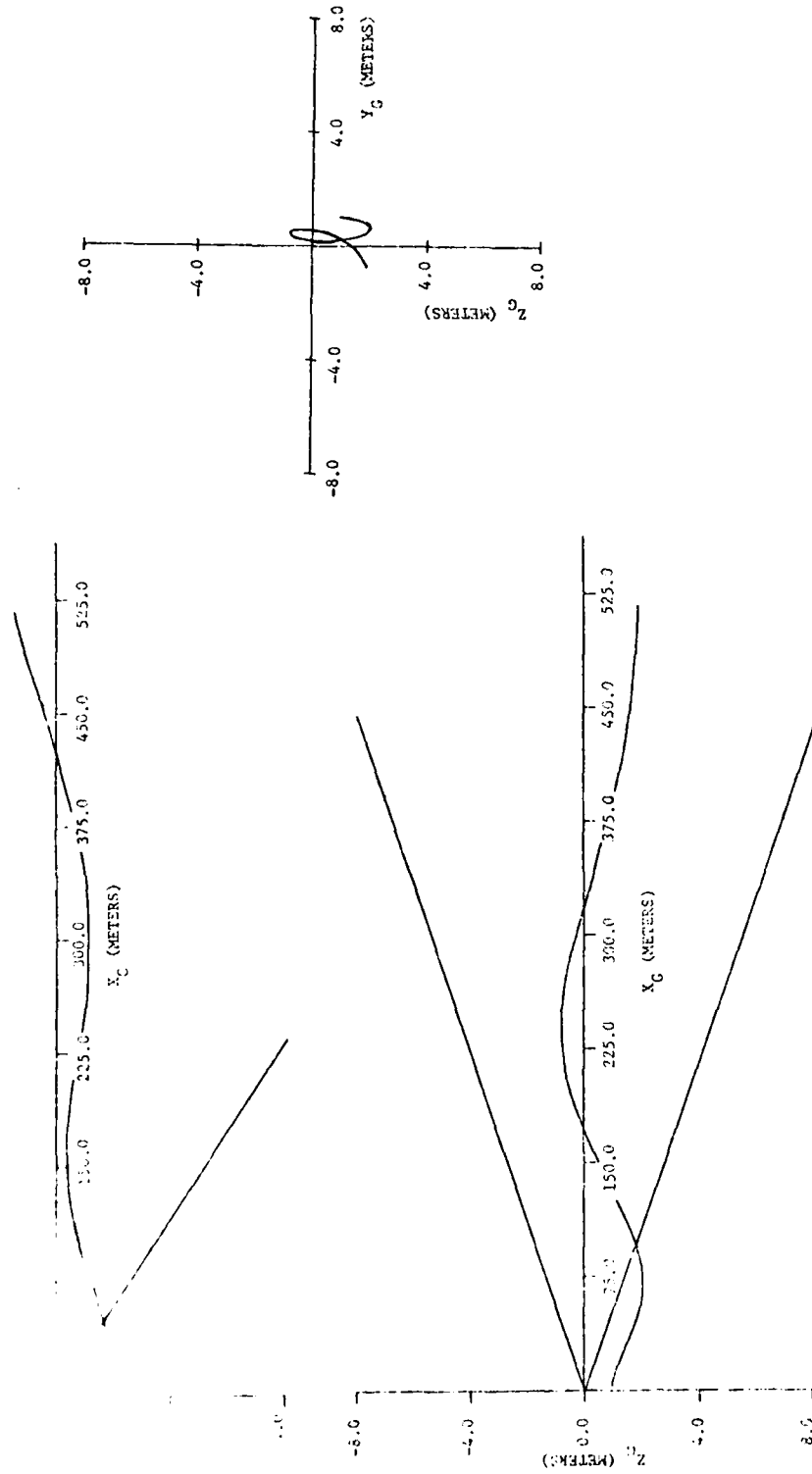


Figure 5-20. Trajectory for Simulation 6.

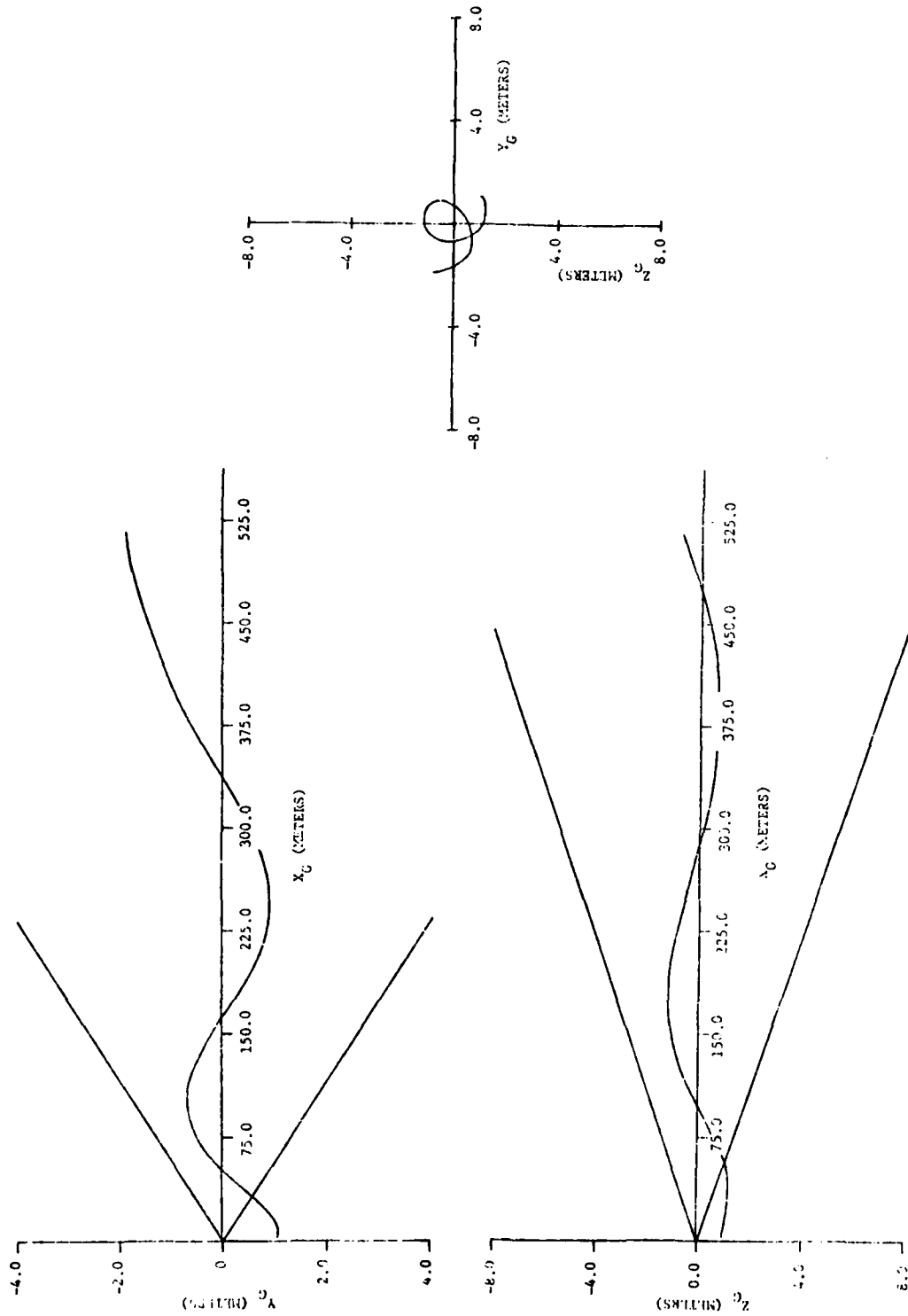


Figure 5-21. Trajectory for Simulation 7.

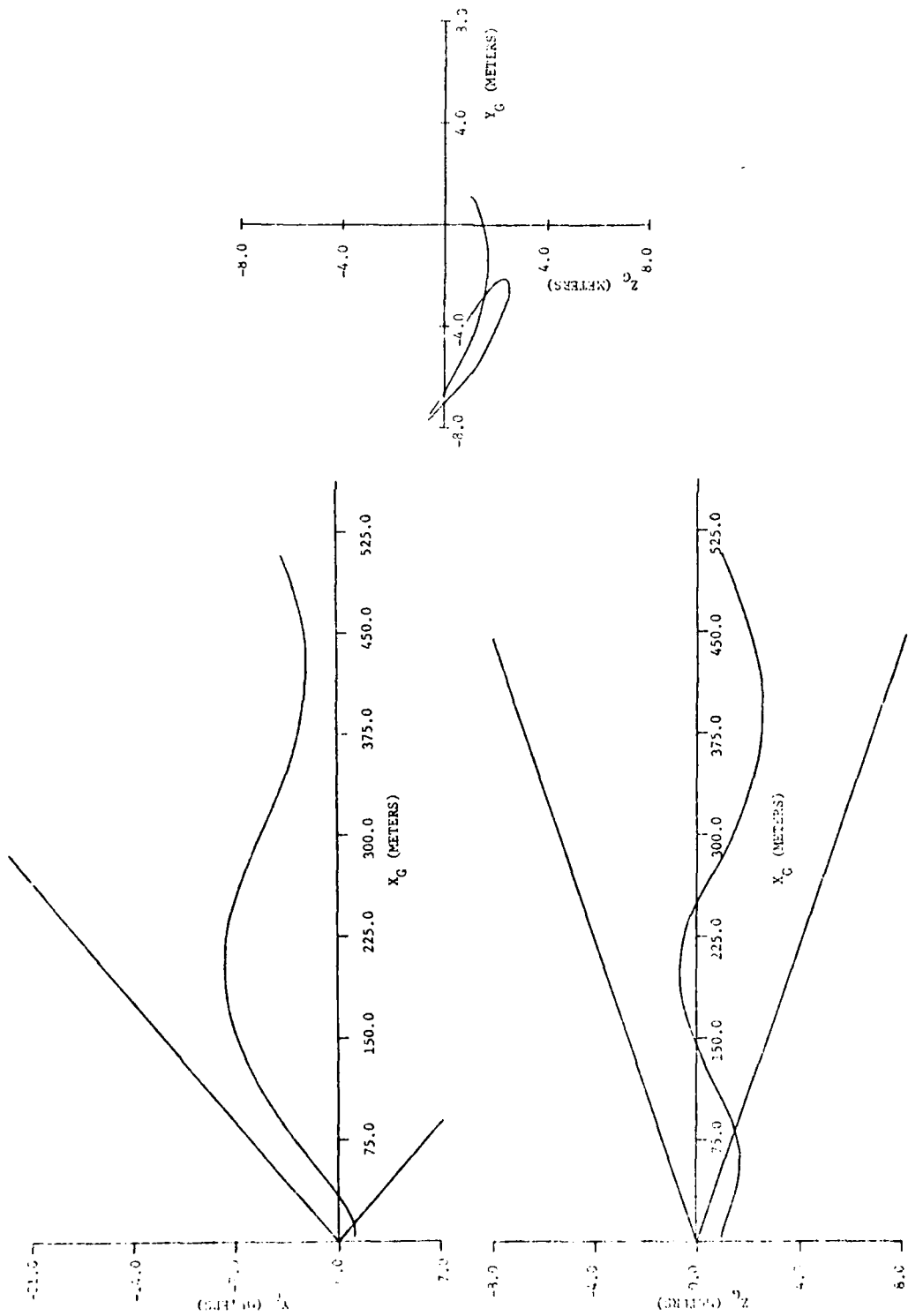


Figure 5-22. Trajectory for Simulation 8.

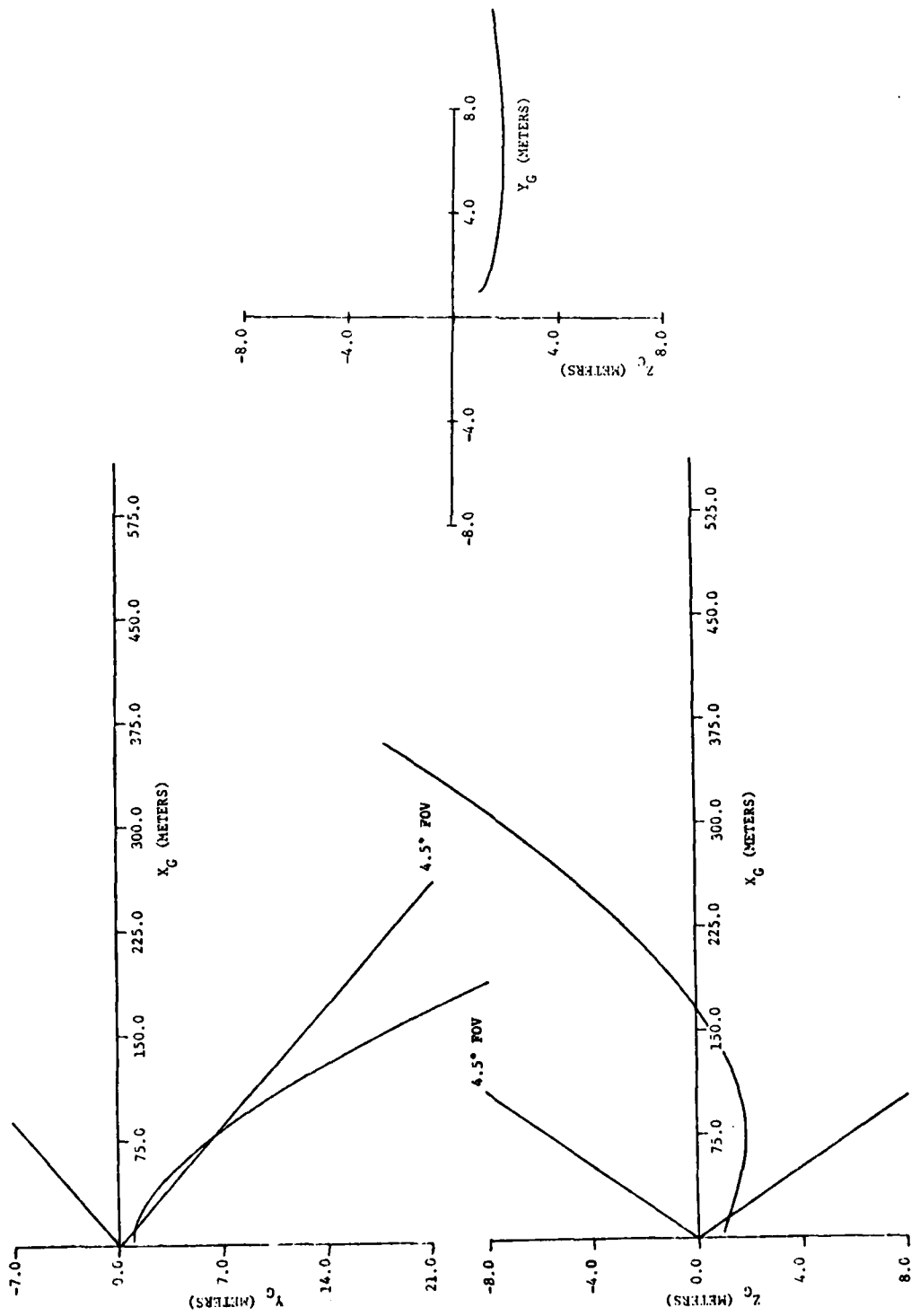


Figure 5-23. Trajectory for Simulation 9.

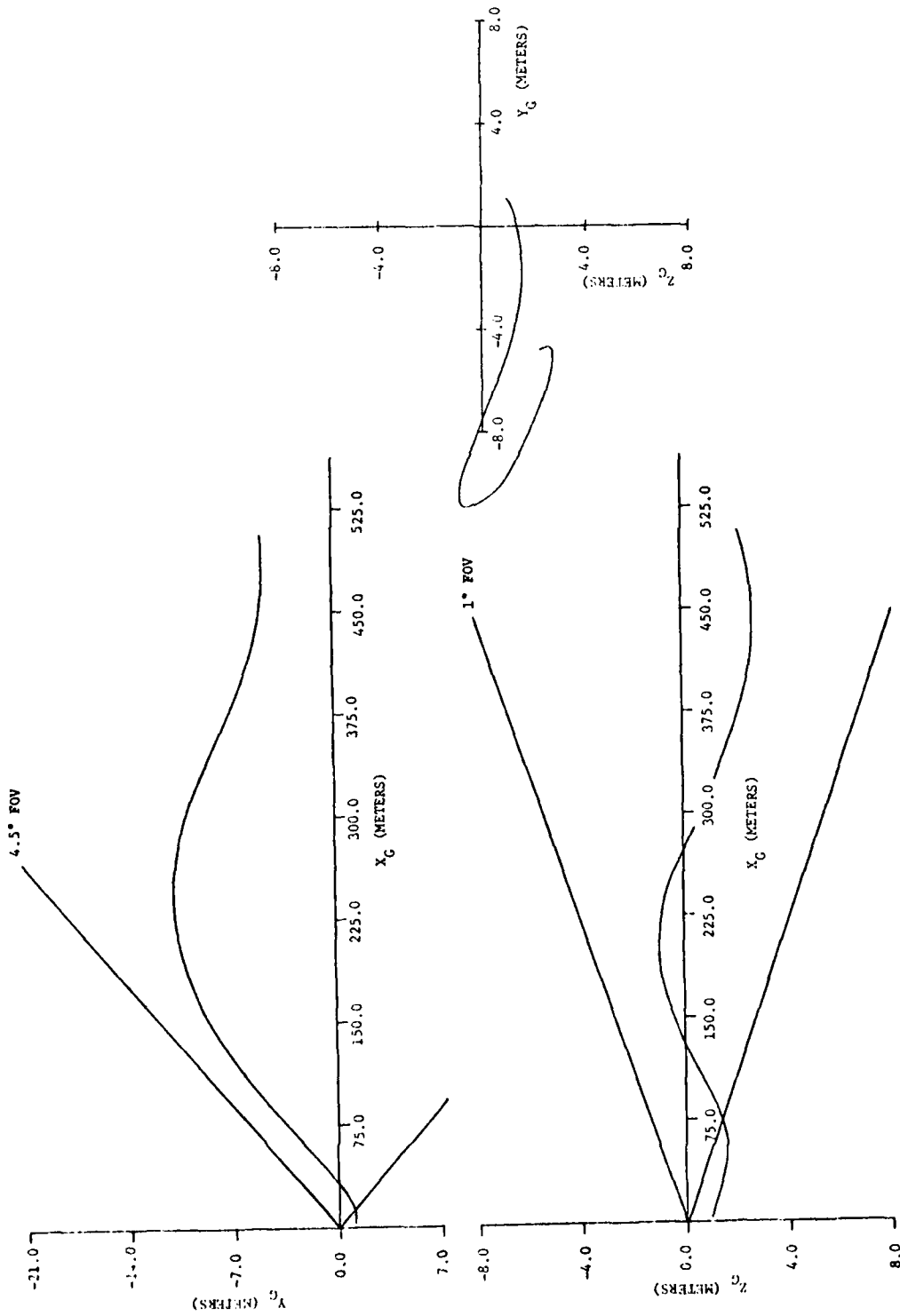


Figure 5-24. Trajectory for Simulation 10.

6. SUMMARY, CONCLUSIONS AND SUGGESTIONS

6.1 Summary

In this report we have described the physical and mathematical models which form the basis for digital simulation codes which may be used in studying the motion of a short-range air defense system rocket during launch and prior to its acquisition by a launcher-based radar. Listings of the FORTRAN IV digital computer codes are also included. The codes were used to simulate a typical system and results of these simulations have been presented and discussed.

6.2 Conclusions

The simulation codes are flexible enough that different systems may be simulated by changing data and/or subroutines. They are not as complex as some codes of a similar nature, since simulation of the operation of all the various subsystems is not attempted. However, many anomalies are modeled and the computer time required for each run is reasonable (around 2.5 minutes on an IBM 370/158 in conjunction with the FORTRAN H compiler.

In regard to the particular system simulated, it was found that the presence of anomalies such as cross-winds, mass imbalance, thrust misalignment, etc., of expected magnitudes did not prevent adequate operation of the system. The only factor which seems to be of major importance is the rate of rotation of the turret, and this is of vital importance only when the rate is large in magnitude (say, $10^\circ/\text{sec}$) and the rocket is fired from a "lagging" tube. This would appear to preclude the use of both rockets against two fast moving targets which appear successively. That is, if a target moving such that a large turret rate is necessary appears, if a rocket

is fired from the leading tube and the rocket guided to impact and, if before the time required for reloading has elapsed, another such target appears, based on the simulations made in this study, the second target cannot be successfully attacked.

6.3 Suggestions

On the basis of experience gained during the course of this study, it is suggested that additional attention be given to determining the effects of the following factors:

1. Induced motions of the rocket due to launcher motions other than the angular rate of the turret about a vertical axis.
2. Unfolding of the wings of the rocket.

It is also suggested that the simulation codes presented herein be modified and/or extended to include mathematical models of the above factors.

REFERENCES

1. Jolly, Alexander C., Dillard, Richard A., Merriman, David B., and Waddle, Franklin M., "Dynamic System Mathematical Models for Roland II," Technical Report RG-76-7, U.S. Army Missile Command, Redstone Arsenal, Alabama, July 1975. (Confidential Document).
2. Cochran, John E., Jr., "Investigation of Factors which Contribute to Mallaunch of Free Rockets," Final Report, U.S. Army Grant DAHC04-75-0034, Engineering Experiment Station, Auburn University, Auburn, AL, January 1976.
3. Mesaoviteh, L., and Bankovkis, J., "Dynamic Characteristics of a Two-Stage Variable Mass Flexible Missile with Internal Flow," NASA CR-2076, June 1972.
4. Miele, A., Flight Mechanics, Vol. I, Theory of Flight Paths, Addison-Wesley Publishing Co., Reading, Mass., 1962, Chapter 3.
5. Beaty, James R., "Development of a Digital Simulation Code for a Short Range Air Defense Missile," Master of Science Thesis, Auburn University, Auburn, Alabama, August 1977.
6. IBM, System 1360 Scientific Subroutine Package (360A-CM-03X) Version III, Programmer's Manual, Fifth Edition, International Business Machines Corporation, White Plains, New York, 1970.

APPENDIX A

LISTING OF MISSILE SIMULATION (MISSIM) PROGRAM

On the following pages is a listing of the missile simulation (MISSIM) program used for the flight phase analysis in the report. The computer code developed is compatible with the FORTRAN G, FORTRAN H, or WATFIV compilers used in conjunction with the IBM 370/158 digital computer available at Auburn University. All computations in the program are done in the double precision mode.

All input into the program is formatted with a generalized (G) field descriptor and is accomplished with punched cards. Each card contains ten data fields which are eight columns wide. The generalized descriptor was chosen because it allows one to input logical, integer or real (external fixed point or floating point) variables using the same format. The entire input necessary to use MISSIM is read in the main program, and is explained below.

NVAL	The number of thrust-time values tabulated for the missile (right-hand justified).
TVECTR	Vector (dimensioned NVAL) of thrust magnitude times (sec).
FVECTR	Vector (dimensioned NVAL) of thrust magnitude values (Nt).
XL	Length of the missile (m).
S	Reference area for aerodynamics (m^2).
D	Reference length for aerodynamics (m).
RP	Outer diameter of the booster motor propellant grain (m).
SRPL	Initial inner diameter of the booster motor propellant grain (m).

RHOP	Booster motor propellant density (kg/m^3).
PGL	Booster motor propellant grain length (m).
ISP	Booster motor special impulse (sec).
GRAV	Gravity magnitude (m/sec^2).
RHO	Atmospheric ambient density (kg/m^3).
TEMP	Atmospheric ambient temperature ($^{\circ}\text{K}$).
GAMMA	Ratio of specific heats for air.
RAIR	Gas Constant for air ($\text{m}^2/\text{s}^2\text{-}^{\circ}\text{K}$).
XP1,XP2	Coordinates of ends of booster motor propellant grain (m).
SLAT	Distance from point A to aft end of missile (m).
RT	Vector from point A to a point on the line of action of the sustainer motor thrust (m).
TIMEV	Five-element vector of times for CG location and moments of inertia values (sec).
VXCG	Five-element vector of CG locations (m).
VIX	Five-element vector of spin moment of inertia values (kg-m^2).
VIY	Five-element vector of pitch moment of inertia values (kg-m^2).
VIZ	Five-element vector of yaw moment of inertia values (kg-m^2).
IXY, IXZ, IYZ	Values of products of inertia. (Used if mass imbalance quantities are not specified) (kg-m^2).
FCMAX	Magnitude of control force (Nt).
DELTA	Angle that the control plane makes with the z-axis (see Fig. 3-4)(rad).
RTUR	Distance from axis of rotation of launcher turret to origin of goniometer system (m).
HANT	Height of radar antenna from origin of inertial system (m).

TBEAM	Half-angle of infrared seeker's field of view (rad).
TMAX	Maximum allowed thrust deflector dwell time (sec).
TDELAY	Delay time from command of control force until control force magnitude equals FCMAX (sec).
MAXRAT	Macimum allowed commanded angular rates (rad/sec).
TBOOST	Time of booster motor burnout (sec).
TBC	Time of sustainer motor ignition (equals earliest time that control is possible)(sec).
EULER	Logical variable defining type of integration routine desired. If Euler is true, integration is Euler; otherwise, fourth-order Runge-Kutta integration is used.
PRTSUP	Logical variable indicating whether excess output is to be suppressed.
PLONLY	Logical variable indicating whether output is to be printed only on the dataset.
H	Integration step size (sec).
NEQ	Number of dependent variables to be integrated (right-hand justified).
NUMB	Number of integration steps before output is printed (right-hand justified).
NUMPPT	Number of integration steps before output is stored for Dataset storage (right-hand justified).
UG,VG,WG	Components of wind velocity in negative direction of inertial axes (m/sec).
TF	Time at end of simulation (sec).
TO	Time at beginning of simulation (sec).
ALY,ALZ	Euler angles which orient the thrust vector (rad).
MU2,MU3	Euler angles which define principal axis rotations. Used to compute dynamic mass imbalance quantities (rad).
X	Vector of initial values of the dependent variables (dimensioned NEQ).

XTV	Initial position vector of target in inertial reference system (m).
VTV	Velocity vector of target in inertial reference system (m/sec).
GAIN1	Control gain K1 (sec ²).
GAIN2	Control gain K2 (sec/m).
GAIN3	Control gain K3 (sec ² /m).
TITLE	Eighty-character alphanumeric vector for the identifying title of the simulation.

Several additional vectors of input data used in subroutine AERO to determine the aerodynamic force and moment on the missile are described in Appendix C.

Output from the program can be obtained in three forms. Which form is obtained depends on input values of PRTSUP and PLONLY (explained above). In the normal form (PRTSUP = FALSE), much output is printed for checking purposes. In the suppressed form (PRTSUP = TRUE and PLONLY = FALSE), only the time and the state variables are printed. In the third form (PRTSUP = TRUE and PLONLY = TRUE) output is stored on DATASET only, and no printed output is obtained.

Comment cards have been provided liberally throughout the program to indicate the most important calculations and to provide the user with additional information.

U U

```

X(12) = -GDELDT*DSIN(X(4)) + SIANDT*DCOS(GDELTA)*DCOS(X(4))
400 CONTINUE
C
C CALL THE EULER/RUNGE KUTTA INTEGRATION ROUTINE
C
C CALL ERKIN.(TO,TF,H,X,NEJ)
C
C STORE THE SOLUTION ON DATASET
C
C CALL SAVPLT
C STOP
C END

```

```

SUBROUTINE SAVPLT
COMMON/COM240/TINVEC(1000),PLVC1(1000),PLVC2(1000),
1PLVC3(1000),PLVC4(1000),PLVC5(1000),PLVC6(1000),PLV
2C7(1000),PLVC8(1000),PLVC9(1000),PLVC10(1000),PLVC11
3(1000),PLVC12(1000),PLVC13(1000),PLVC14(1000),PLVC
415(1000),PLVC16(1000),PLVC17(1000),PLVC18(1000),NN
COMMON/COM100/TITLE(20)
C
C WRITE THE TITLE OF THE RUN, ALONG WITH THE SOLUTION
C OF THE SIMULATION, ON DATASET IN UNFORMATTED FORM
C USING UNIT NUMBER 11.
C
C WRITE(11) (TITLE(I),I=1,20)
C WRITE(11) NN
C DO 100 I=1,NN
C WRITE(11) TINVEC(I),PLVC1(I),PLVC2(I),PLVC3(I),PLVC4(I)
C 1,PLVC5(I),PLVC6(I),PLVC7(I),PLVC8(I),PLVC9(I),PLVC10(I)
C 2,PLVC11(I),PLVC12(I),PLVC13(I),PLVC14(I),PLVC15(I),PLVC
316(I),PLVC17(I),PLVC18(I)
C 100 CONTINUE
C RETURN
C END

```

```

SUBROUTINE ERKINT(TO,TF,HIN,X,N)
IMPLICIT REAL*8(A-H,K,O-Z)
LOGICAL EULER
DIMENSION K1(13),K2(13),K3(13),K4(13),X(13),DELTA X(
113),XPRIME(13),XDOT(13),X LAST(13)
COMMON/COM150/EULER
F=TO
IF (F0 .EQ. TF) GO TO 1300
SIGN = (DABS(TF-TO))/(TF-TO)
IF (SIGN .LT. 0. .AND. HIN .GT. 0.) GO TO 1700
IF (HIN .EQ. 0.) GO TO 1500
100 CONTINUE
C
C SAVE THE VALUE OF X(I) AT THIS STEP.
C
H=HIN
DO 200 I=1,N
200 X LAST(I)=X(I)
CALL FUNCT(T,X,XDOT)
C
C OUTPUT THE VALUE OF T,X, AND XDOT AT THIS STEP.
C
CALL OUTPUT(T,X,XDOT,TO,TF,H)
C
C IF EULER IS TRUE, THIS IS EULER INTEGRATION
C SO SKIP THE RUNGE KUTTA SECTION
C
IF (EULER) GO TO 700
C
C BEGIN THE SECTION TO COMPUTE THE K'S .
C
T PRIME=T+.5*H
DO 300 I=1,N
K1(I)=H*XDOT(I)
300 X PRIME(I)=X(I)+K1(I)*.5
CALL FUNCT(T PRIME,X PRIME,XDOT)
DO 400 I=1,N
K2(I)=H*XDOT(I)
400 X PRIME(I)=X(I)+.5*K2(I)
CALL FUNCT(T PRIME,X PRIME,XDOT)
DO 500 I=1,N
K3(I)=H*XDOT(I)
500 X PRIME(I)=X(I)+K3(I)
T PRIME=T+H
CALL FUNCT(T PRIME,X PRIME,XDOT)
DO 600 I=1,N
K4(I)=H*XDOT(I)
C
C NOW THAT THE K'S HAVE BEEN CALCULATED, COMPUTE THE
C VALUE OF DELTA X FROM THE WEIGHTED AVERAGE FORMULA.
C

```



```

C 600 DELTA X(I) = (K1(I) + 2*K2(I) + 2*K3(I) + K4(I))/6.
      GO TO 900
700 CONTINUE
      DO 800 I=1,N
800 DELTA X(I) = H*XDOT(I)
C
C      ADD DELTA X TO X TO GET ITS NEW VALUE FOR THE NEXT STEP.
900 CONTINUE
      DO 1000 I=1,N
1000 X(I) = X(I) + DELTA X(I)
C
C      CHECK TO SEE IF T IS STILL IN THE SPECIFIED INTEGRATION
C      INTERVAL. IF IT IS, CONTINUE INTEGRATING. IF IT IS
C      NOT, STOP HERE.
      IF (SIGN -GT. 0. .AND. T -LT. TF) GO TO 1200
      IF (SIGN -LT. 0. .AND. T -GT. TF) GO TO 1200
      T=H-T
      DO 1100 I=1,N
      IM1=I-1
1100 X(I) = X LAST(I) + ((TF-TMH)/(T-TMH)) * (X(I) - X LAST(I))
      I=TF
C
C      OUTPUT THE VALUES FOR T, X, AND XDOT FOR THE FINAL
C      INTERVAL.
      CALL OUTPUT(T,X,XDOT,TO,TF,H)
      GO TO 1500
1200 CONTINUE
C
C      SINCE T IS STILL IN THE SPECIFIED INTEGRATION INTERVAL
C      GO TO 130 AND CONTINUE INTEGRATING.
      T=T+H
      GO TO 100
C
C      THIS SECTION PRINTS OUT THE APPROPRIATE ERROR MESSAGES
C      IF THE INPUT DATA FOR THE RUNGE-KUTTA ARE NOT FEASIBLE.
C
1300 WRITE(6,1400) TO
1400 FORMAT('H1,*****ERROR: INTERVAL SIZE IS',
1, 'ZERO (TO=TF= ',F8.0,').*****')
      GO TO 1900
1500 WRITE(6,1600)
1600 FORMAT('H1,*****ERROR: STEP SIZE IS',
1, 'ZERO.*****')
      GO TO 1900
1700 WRITE(6,1800)

```

```

1800 FORMAT('H1,*****ERROR: SIGN OF THE STEP',
1, 'SIZE IS NOT COMPATIBLE WITH THE INTEGRATION INTERVAL.',
2, '*****')
1900 CONTINUE
      RETURN
      END
C
C      SUBROUTINE OUTPUT(TIME,X,XDOT,IO,TF,H)
      IMPLICIT REAL*H(A-H,O-Z)
      REAL*4 TIMEV,PLVC1,PLVC2,PLVC3,PLVC4,PLVC5,PLVC6,
1 PLVC7,PLVC8,PLVC9,PLVC10,PLVC11,PLVC12,PLVC13,PLVC14,
2 PLVC15,PLVC16,PLVC17,PLVC18,SNGL
      LOGICAL PRISUP,PLONLY
      REAL*8 MCV,MZ,MX,MXAT
      REAL*8 JDCOEF,IT,LT,IS
      DIMENSION X(13),XDOT(13)
      COMMON/COM230/XG,XCP,STHAR,ALPDEG,BETDEG,UGUST,VGUST,
1 WJUST,VR
      COMMON/COM50/ICMAX,DELTA,GRAV,CMA,CZA,CHQ,UDOT,DIA,
1 RHOS,IT,IS,L2,JDCOEF,TMAX,IDELAY,MXAT,TBOOST,TBC
      COMMON/COM180/XG(3),VG(3),DEGDEL,DEGLAN,ENR,ENR
      COMMON/COM190/DELTA,DELTA,ANGV,ANGZ,PHIPD,SGN,FCY,PCZ,
1 MCV,MZ,XK1,XK2,XK3
      COMMON/COM220/R,XDOT,EPST,EPST,EPST,EPST,EPST,D,DHAX
      COMMON/COM240/TINVEC(1000),PLVC1(1000),PLVC2(1000),
1 PLVC3(1000),PLVC4(1000),PLVC5(1000),PLVC6(1000),PLV
2 C7(1000),PLVC8(1000),PLVC9(1000),PLVC10(1000),PLVC
3 11(1000),PLVC12(1000),PLVC13(1000),PLVC14(1000),PLVC
4 15(1000),PLVC16(1000),PLVC17(1000),PLVC18(1000),NN
      COMMON/COM90/NUMB,NUMPPT
      COMMON/COM210/ONRC,ONRC,DLTY,DLTY
      COMMON/COM140/PLONLY,PRISUP
      RAD=57.29577951D0
      IF(DABS(TIME-TO) -GT. 1.D-5) GO TO 100
      NN=0
      NUMBPL=(TF-TO)/H/NUMPPT
      NSTEP=NUMB
      PRINT 1300,XK1,XK2,XK3
      GO TO 200
100 IF(NPSTEP -GE. NUMBPL) GO TO 200
      NPSTEP=NPSTEP+1
      GO TO 300
200 NN=NN+1

```

STORE THE SOLUTION VALUES AT THIS STEP IN ARRAYS
FOR DATASET STORAGE LATER.

BEST AVAILABLE COPY

```

ZINVEC(NN)=SNGL(TIME)
PLVC1(NN)=SNGL(XG(1))
PLVC2(NN)=SNGL(XG(2))
PLVC3(NN)=SNGL(XG(3))
PLVC4(NN)=SNGL(X(5)*RAD)
PLVC5(NN)=SNGL(X(6)*RAD)
PLVC6(NN)=SNGL(X(8)*RAD)
PLVC7(NN)=SNGL(X(9)*RAD)
PLVC8(NN)=SNGL(X(11))
PLVC9(NN)=SNGL(X(12))
PLVC10(NN)=SNGL(QNR)
PLVC11(NN)=SNGL(RNR)
PLVC12(NN)=SNGL(VNRC)
PLVC13(NN)=SNGL(RNRC)
PLVC14(NN)=SNGL(DLTY)
PLVC15(NN)=SNGL(DLTZ)
PLVC16(NN)=0.
PLVC17(NN)=0.
PLVC18(NN)=0.
NPSTEP=1
300 IF(NSTEP-GE. NUMB)GO TO 400
NSTEP=NSTEP+1
RETURN
+00 ANG4=X(4)*RAD
ANG5=X(5)*RAD
ANG6=X(6)*RAD
ALPDEG=X(8)*RAD
BETDEG=X(9)*RAD
W=X(7)*DTAN(X(8))
VEL=DSQRT(X(7)**2+W**2)
V=VEL*DTAN(X(9))
NSTEP=1

IF(PRTSUP IS TRUE, SOME EXTRA OUTPUT IS TO BE
SUPPRESSED, SO SKIP THIS SECTION.

IF(PRTSUP)GO TO 600
PRINT 800,TIME,X(13),XG,XCP,STAR,DEGDEL,DEGLAN,A
1LPDEG,BETDEG
PRINT 900,CZA,CMA,CAD,UDOF,IT,IS,LI,JDCOEF,A
PRINT 1000,R,ROOF,D,DHAX,X(1),X(2),X(3)
PRINT 1100,DELTZ,ANGZ,ANGZ,PHIND,SGN,FCY,PCZ,
IMCY,MCZ
500 PRINT 1400,(XDOT(I),I=1,14)
PRINT 1500,X(1),XG(1),ANG4,X(10),X(7),VG(1),X(2),XG(2)
1,ANG5,X(11),JNR,V,VG(2),X(3),XG(3),ANG6,X(12),RNR,W,
2VG(3)
RETURN

IF PLONLY IS TRUE, ONLY DATA SET STORABLE OF THE
SOLUTION IS DESIRED, SO NO OUTPUT IS TO BE PRINTED
ON PAPER.

```

```

600 IF(PLONLY)GO TO 700
PRINT 1200,TIME
GO TO 500
700 RETURN

C
800 FORMAT(1H,///,3X,TIME=,F7.4,3X,MASS=,G12.5,3X,
1,XCG=,G12.5,3X,XCP=,G12.5,3X,S1. MAR. =,G12.5,/,
21H,2X,DELTA G=,G12.5,3X,LAMBDA G=,G12.5,3X,
3,ALPHA=,G12.5,3X,BETA=,G12.5)
900 FORMAT(1H,2X,CZA=,G12.5,3X,CMA=,G12.5,3X,CAD=,
1G12.5,3X,UDOF=,G12.5,3X,IT=,G12.5,/,1H,2X,IS=,
2G12.5,3X,LT=,G12.5,3X,JDCOEF=,G12.5,3X,IM1. STEP=,
3,G12.5)
1000 FORMAT(1H,2X,R=,G12.5,3X,RDOF=,G12.5,3X,D=,G12
1.5,3X,DHAX=,G12.5,/,1H,2X,K1=,G12.5,3X,K2=,G12
2.5,3X,K3=,G12.5)
1100 FORMAT(1H,2X,DELTZ=,G12.5,3X,DELTY=,G12.5,3X,
1,ANGY=,G12.5,3X,ANGZ=,G12.5,3X,PHIPD=,G12.5,/,1H
2,2X,SIGN=,G12.5,3X,FCYNR=,G12.5,3X,PCZNR=,G12
3.5,3X,HCYNR=,G12.5,3X,HCZNR=,G12.5)
1200 FORMAT(1H,///,3X,TIME=,G12.5)
1300 FORMAT(1H,3X,K1=,G12.5,3X,K20=,G12.5,3X,K30=,
1G12.5)
1400 FORMAT(1H,2X,7(G12.5,5X))
1500 FORMAT(1H,/,3X,XV=,G12.5,3X,XG=,G12.5,3X,PHI =,
1,G12.5,3X,P=,G12.5,3X,PNR=0.00
22.5,3X,UG=,G12.5,/,1H,2X,YV=,G12.5,3X,YG=,G12
3.5,3X,THETA=,G12.5,3X,Q=,G12.5,3X,QNR=,G12.5,3X
4,V=,G12.5,3X,VG=,G12.5,3X,/,1H,2X,ZV=,G12.5,
53X,ZG=,G12.5,3X,PSI =,G12.5,3X,R=,G12.5,3X,
6,RNR=,G12.5,3X,W=,G12.5,3X,WG=,G12.5)
C
END

```

```

SUBROUTINE FUNCT(TIME,X,XDOT)
  IMPLICIT REAL*4 (A-H,O-Z)
  REAL*4 ADOT,MASS,INERTA,ININV,IGTOTM,LBV,LBVINV,IX,
  IY,IY2,MY,MY2,IZ,IZ2,ISP,LENGTH,MG,MJD,JDCOEF,
  2492,MU3,MAXRAT
  DIMENSION X(13),XDOT(13),TVECTR(50),FVECI(50),F(3),
  1FA(3),FT(3),F3(3),FC(3),MA(3),MI(3),MC(3),DOmega(3),
  2VTRANS(3),MJD(3),IDOTM(3),ANGACL(3),AA(3),RT(3),
  3OMEGAX(3),VB(3),ANG(3),OMEGAT(3,3),INERTA(3,3),ININV
  4(3,3),JNEGA(3),VV(24),VEC1(3),VEC2(3),RTSTOR(3),RT TILD
  5(3,3),LBV(3,3),LBVINV(3,3)
  COMMON/COM120/ISP,IXY,IXZ,IY2,PGL,U6,V6,W6,LENGTH,XP1,
  1XP2,GRAV,SLAT,RTSTOR,NVAL
  COMMON/COM10/FA,MA
  COMMON/COM40/FVECTR,TVECTR
  COMMON/COM20/ALFAP,ALFAZ,MU2,MU3
  COMMON/COM50/FCMAZ,DELTA,GRV,CMA,CZA,CMA,UDOT,DIA,
  1RHUS,IYIX,KL1,JDCOEF,TMAX,TDELAY,MAXRAT,TBOOST,TBC
  COMMON/COM160/FC,MC
  COMMON/COM110/TIMEV(5),VIX(5),VIY(5),VIZ(5),VXCG(5)
  COMMON/COM230/XCG,XCP,STHAP,ALPHA,BETA,UGUST,VGUST,
  1UGUST,VR
  PI=3.141592654D0
  THOP1=2.*PI
  X(4)=DMOD(X(4),THOP1)
  PHIEG=X(4)*57.29577951D0
  MASS=X(13)
  DO 100 I=1,3
    JNEGA(I)=X(I+9)
  100 CONTINUE
  ANG(1)=X(4)
  ANG(2)=X(5)
  ANG(3)=X(6)
  DO 200 I=1,24
    VV(I)=0.D0
  200 VV(I)=0.D0
  DO 300 I=1,3
    VV(I+3)=ANG(I)
  300 VV(I+3)=ANG(I)
  CALL CHAT(VV,LBV,LBVINV,SPSL,CPSL,STHETA,CTHETA,SPHL,
  1CPHL)
  C
  C TRANSFORM WIND VELOCITY FROM INERTIAL TO BODY SYSTEM
  C
  VEC1(1)=UG
  VEC1(2)=VG
  VEC1(3)=WG
  CALL MATRV(LBV,VEC1,VEC2)
  UGUST=VEC2(1)
  VGUST=VEC2(2)
  WGUST=VEC2(3)

```

```

C      MJD(1)=0.D0
C      MJD(2)=-2.*JD COEPCX(11)
C      MJD(3)=-2.*JD COEPCX(12)
C
C      INTERPOLATE TO FIND THE VALUES OF MOMENTS OF INERTIA
C      AT THIS TIME.
C
C      CALL INTERP(TIME,IX,TIMEV,VIX,5,NSTOP)
C      IF(NSTOP -LT. 0) STOP 110
C      CALL INTERP(TIME,IY,TIMEV,VIY,5,NSTOP)
C      IF(NSTOP -LT. 0) STOP 120
C      CALL INTERP(TIME,IZ,TIMEV,VIZ,5,NSTOP)
C      IF(NSTOP -LT. 0) STOP 140
C
C      INERTA(1,1)=IX
C      INERTA(1,2)=-IXY
C      INERTA(1,3)=-IXZ
C      INERTA(2,1)=-IXY
C      INERTA(2,2)=IY
C      INERTA(2,3)=-IYZ
C      INERTA(3,1)=-IXZ
C      INERTA(3,2)=-IYZ
C      INERTA(3,3)=IZ
C
C      COMPUTE DYNAMIC IMBALANCE QUANTITIES.
C
C      INERTA(1,2)=-.933*(INERTA(2,2)-INERTA(1,1))
C      INERTA(1,3)=.602*(INERTA(3,3)-INERTA(1,1))
C      INERTA(2,1)=INERTA(1,2)
C      INERTA(3,1)=INERTA(1,3)
C
C      COMPUTE THE WEIGHT COMPONENTS IN BODY AXES
C
C      MG=MASS*GRAV
C      PG(1)=-MG*STHETA
C      PG(2)=MG*CTHETA*SPHI
C      PG(3)=MG*CTHETA*CPHI
C
C      CALL THE CONTROL SUBROUTINE TO OBTAIN THE CONTROL
C      FORCES AND MOMENTS.
C
C      CALL CTRL(TIME,X,XDOT)
C      DO 500 I=1,J
C        F(I)=FA(I)+FG(I)+FT(I)+FC(I)
C        CALL TILDE(OMEGA,ONEGAT)
C        CALL MATRX(OMEGAT,VB,OMEGAV)
C        UDOT=-OMEGXV(1)+F(1)/MASS
C        VDOT=-OMEGXV(2)+F(2)/MASS
C        WDOT=-OMEGXV(3)+F(3)/MASS
C        XDOT(7)=UDOT
C        CONST1=X(7)/VEL/VEL

```

```

      JS=0.5*RH0*V*V*VRT*V
      ALPHA1=DATA2(4T,UE)
      CONST=D583T((X(7)+UGUST)**2*(W+UGUST)**.)
      BETAI=DATA2(VF,CONST)
      ALPHABEG=ALPHAT*57.2957795100
      BETDEG=BETAT*57.2957795100
      ITANGL=D583T(BETAI*BETAT+ALPHAT*ALPHAT)

      INTERPOLATE FOR CNA,XCP,CNO,AND CQ.

      IF(ITANGL.GT..0698132)GO TO 100
      CALL INTERP(MACH,CNA,MACHVC,CNAVCL,27,NSTOP)
      IF(NSTOP.LT.0)STOP 10
      GO TO 200

100 CONTINUE
      CALL INTERP(MACH,CNA,MACHVC,CNAVCL,27,NSTOP)
      IF(NSTOP.LT.0)STOP 20
      GO TO 200

200 CONTINUE
      CNA=CNA
      CVBETA=CNA
      IF(LMACS.LT.XVEC(1))LMXCG=XVEC(1)
      IF(LMACS.GT.XVEC(3))LMXCG=XVEC(3)
      IF(TIME.GE..500)GO TO 300
      CALL INTERP(LMXCG,CNQ,XVEC,CNQVC1,3,NSTOP)
      GO TO 400

300 CALL INTERP(LMXCG,CNQ,XVEC,CNQVC2,3,NSTOP)
400 CONTINUE
      CNR=CNQ
      CALL INTERP(MACH,CQ,MACHVC,CQVEC,27,NSTOP)
      IF(NSTOP.LT.0)STOP 30
      CALL INTERP(MACH,CNO,MACHVC,CNOVEC,27,NSTOP)
      IF(NSTOP.LT.0)STOP 40
      IF(ITANGL.LE..0698132)CNO=0.00
      CY0=CNO
      LR=58.8235*(TIME-.8)
      IF(TIME.LT.0.8)LR=0.00
      IF(TIME.GT.2.5)LR=100.00
      CALL INTERP(LR,BR2,LRVEC,BRVFC1,22,NSTOP)
      IF(NSTOP.LT.0)STOP 50

      WITH THE VALUES FOR MACH NUMBER AND LR2 FOUND ABOVE,
      NOW GO TO THE P1 MATRIX (THE TABLE OF VALUES FOR P1
      VERSUS MACH NO. AND BR2) AND INTERPOLATE TO FIND P1.

      DO 500 I=1,10
      IF(BRVEC2(I).GT.BR2)GO TO 600
500 CONTINUE
600 CONTINUE
      INI=I-1
      DO 700 J=1,6
      VECTRI(J)=PIHAT(LMI,J)
700 VECTR2(J)=PIHAT(I,J)
      M=MACH

```

```

      IF(MACH.LT.MVECTR(1))M=MVECTR(1)
      IF(MACH.GT.MVECTR(6))M=MVECTR(6)
      CALL INTERP(4,P11,MVECTR,VECTR1,6,NSTOP)
      IF(NSTOP.LT.0)STOP 60
      CALL INTERP(4,P12,MVECTR,VECTR2,6,NSTOP)
      IF(NSTOP.LT.0)STOP 70
      P1=P11+((P12-P11)/(LRVEC2(I)-BRVEC2(I)))*(BR2-BRVEC
      12(LMI))

      NOW THAT P1 HAS BEEN FOUND, THE VALUE FOR P2 CAN BE
      FOUND BY INTERPOLATING THE P2 VECTOR FOR THE VALUE OF
      ITANGL.

      ANGLIT=ITANGL*57.2957795100
      CALL INTERP(ANGLIT,P2,ITVECT,P2VEC,9,NSTOP)
      IF(NSTOP.LT.0)STOP 100

      NOW THAT P1 AND P2 HAVE BEEN FOUND, THE MISSILE'S
      CENTER OF PRESSURE, MEASURED FROM THE NOSE, CAN
      BE FOUND.

      L1=(P1+P2)*.001
      XCP=L-L1
      XPCP=XCP
      STARH=(XCP-XCG)/DIA
      CMA=STARH*CZA
      CY=CXBETA*BETAT+CYO
      CN=CNA*ALPHAT+CNO

      COMPUTE THE AERODYNAMIC FORCES AND MOMENTS

      PA(1)=-QS*CD
      PA(2)=QS*CY
      PA(3)=-QS*CN
      IF(MACH.GT.1.0)GO TO 800
      CLP=-10.8009-1.15415*ITANGL+.168189*ITANGL*ITANGL
      1-8.37143E-03*ITANGL**3-3.73996E-05*ITANGL**4
      GO TO 900
800 CLP=-9.40571-1.20658*ITANGL+.057145*ITANGL*ITANGL
900 CONTINUE
      CLNU=544.88286D0
      NU=-.00541D0
      LO=0.00
      IF(VRT.LT.1.0-5)GO TO 1000
      MA(1)=QS*DIA*(CLP*X(10)+.5*DIA/VRT*CLNU*NU)+LO
      MA(1)=0.00
      MA(2)=QS*DIA*(CNQ*X(11)+.5*DIA/VRT*CN*(XCG-XCP))
      MA(3)=QS*DIA*(CNR*X(12)+.5*DIA/VRT*CY*(XCP-XCG))
      GO TO 1100
1000 MA(1)=0.00
      MA(2)=0.00
      MA(3)=0.00
1100 CONTINUE
      RETURN
      END

```

100

```

SUBROUTINE CONTRL(TIME,X,XDOT)
IMPLICIT REAL*8(A-H,M,Q-Z)
LOGICAL CNTR ON,AUX1,AUX2,AUX3,AUX4,AUX5
REAL*8 IT,IS,LI,K1,K2,K3,NUMB,JDCOEF,K20,K30
DIMENSION FC(3),MC(3),X(13),XDOT(13)
COMMON/COM50/FCMA,DELTA,GRAY,CHA,CZA,CNG,UDOT,DIA,RHO
15,IT,IS,LI,JDCOEF,TMAX,TDELAY,MAXRAF,TBOOST,TBC
COMMON/COM60/T0,TF,HIN
COMMON/COM80/XK1,K20,K30
COMMON/COM160/FC,MC
COMMON/COM170/ZG,YG,ZGDOT,YGDOT
COMMON/COM180/XG(3),VG(3),DEGDEL,DEGLAM,CNR,RNR
COMMON/COM190/DELTY,DELTZ,ANGY,ANGZ,PHIPD,SGN,FCYNR,
1FCZNR,MCYNR,MCZNR,K1,K2,K3
COMMON/COM200/D,DMAX
COMMON/COM210/JNRC,RNRC,DLTY,DLTZ
SIGN(X)=DSIGN(1,D0,X)
K1=XK1
PI=3.141592654D0
TWOPI=2.D0*PI
PHI=X(4)
U=X(7)
MASS=X(13)
OMEGA=X(10)
TDEL2=2.D0*TDELAY
Q=X(11)
R=X(12)
JNB=J*DCOS(PHI)-B*DSIN(PHI)
RNR=Q*DSIN(PHI)+R*DCOS(PHI)
JN=MASS*U
PHIPD=X(4)+DELTA
PHIPD=DRND(PHIPD,TWOPI)
SGN=0.

C SET THE AXIAL FORCE AND THE ROLLING MOMENT DUE TO THE
C CONTROL DEFLECTORS EQUAL TO ZERO, SINCE THE DESIGN
C OF THE DEFLECTORS DOES NOT ALLOW FOR EITHER OF
C THESE REACTIONS.
C
FC(1)=0.D0
MC(1)=0.D0
IF(TIME.LE. IBC) CNTR ON=.FALSE.
K2=K20*TBC/TIME
K3=K30*TBC/TIME
IF(TIME.GT. IBOOST) K2=K20*TBC/TBOOST
IF(TIME.GT. TBOOST) K3=K30*TBC/TBOOST

C CALL TRACKER ROUTINE TO RECEIVE TELEMETRY COMMANDS
C
CALL TRACKR(TIME,X,XDOT,2)

```

```

IF CNTR ON IS TRUE, SKIP THE SECTION TO COMPUTE
DELTY AND DELTZ

```

```

IF (CNTR ON) GO TO 300

```

```

IF TIME IS LESS THAN THE TIME OF SUSTAINER MOTOR
IGNITION, CONTROL IS NOT POSSIBLE

```

```

IF (TIME.LT. TBC) GO TO 400

```

```

CHECK AND SEE IF WE HAVE ALREADY CONTROLLED DURING
THIS CONTROL PERIOD. IF SO, CONTROL IS NOT
POSSIBLE. IF NOT, CHECK TO SEE IF CONTROL IS
POSSIBLE THIS STEP.

```

```

ANGMAX=THAX*OMEGA

```

```

A1=-5*(PI-ANGMAX)

```

```

A2=PI-.5*ANGMAX

```

```

A3=-5*(3.*PI-ANGMAX)

```

```

A4=THOPI-.5*ANGMAX

```

```

AJX1=PHIPD.GT. 0. .AND. PHIPD.LT. A1

```

```

CONST=.5*PI

```

```

AJX2=PHIPD.GT. CONST.AND. PHIPD.LT. A2

```

```

AJX3=PHIPD.GT. PI.AND. PHIPD.LT. A3

```

```

CONST=1.5*PI

```

```

AJX4=PHIPD.GT. CONST.AND. PHIPD.LT. A4

```

```

AJX5=AUX1.OR. AUX2.OR. AUX3.OR. AUX4

```

```

IF (AJX5) GO TO 400

```

```

COMPUTE DELTA T, THE TIME INCREMENT FOR THE THRUST
DEFLECTORS TO REMAIN ENGAGED, FOR BOTH THE LONGI-
TUDINAL AND LATERAL EQUATIONS.

```

```

IF THE FALL OF THE MISSILE IS OUTSIDE THE TRACKER
SEAM CONE, NO GUIDANCE COMMANDS ARE SENT, SO ONLY
RATE STABILIZATION COMMANDS ARE POSSIBLE
(I.E. -- COMMANDED PITCH AND YAW RATES ARE ZERO).

```

```

IF (J.GT. DMAX) GO TO 100

```

```

JNRC=K2/K1*ZG+K3/K1*ZGDOT

```

```

RNRC=K2/K1*YG+K3/K1*YGDOT

```

```

IF (DABS(JNRC).GT. MAXRAF) JNRC=MAXRAF*SIGN(JNRC)

```

```

IF (DABS(RNRC).GT. MAXRAF) RNRC=MAXRAF*SIGN(RNRC)

```

```

GO TO 200

```

```
100 CONTINUE

```

```

JNRC=0.

```

```

RNRC=0.

```

```
200 CONTINUE

```

```

DELT2=-K1*(XNR-XNR0)
DELT2=-K1*(XNR-XNR0)
FIN=TIME
IF(DABS(DELT2) .GT. TMAX) DELTY=TMAX*SIGN(DELT2)
IF(DABS(DELT2) .GT. TMAX) DELTZ=TMAX*SIGN(DELT2)
C
C CHECK TO SEE IF THE MISSILE IS IN THE CORRECT ROLL
C ATTITUDE FOR LONGITUDINAL OR LATERAL CONTROL.
C
ANGY=DABS(OMEGA*DELT2)
ANGZ=DABS(OMEGA*DELT2)
AA=ANGY*.500
BB=PI-AA
CC=.5*(PI-ANGZ)
DD=.5*(PI+ANGZ)
IF(DABS(DELT2) .LT. TDEL2 .AND. DABS(DELT2) .LT.
TDEL2) GO TO 500
300 CONTINUE
IF(PHIPD .GT. PI) PHIPD=PI-PI
ASSPPD=DABS(PHIPD)
IF(ABSPPD .GE. BR .OR. ASSPPD .LE. AA) GO TO 600
IF(ABSPPD .GE. CC .AND. ASSPPD .LE. DD) GO TO 800
C
C NO CONTROL THIS STEP
C
GO TO 450
400 XNR=0.
XNR0=0.
450 DELTY=0.
DELTZ=0.
ANGY=0.
ANGZ=0.
FCYNR=0.
FCZNR=0.
CNR ON = .FALSE.
DLTY=0.
DLTZ=0.
GO TO 1000
C
C LONGITUDINAL CONTROL
C
600 IF(DABS(DELT2) .LT. TDEL2) GO TO 500
SGN=SIGN(DELT2)*SIGN(DCOS(PHIPD))
IF(DABS(SGN) .LT. 1.D-6) SGN=DABS(DELT2)/DELT2
TOUT=TIN+DABS(DELT2)
T1=TIN+TDELAY
T2=TOUT-TDELAY
CONST=1.D0
IF(TDELAY .LE. 0.) GO TO 700
IF(TIME .LT. T1) CONST=1./TDELAY*(TIME-TIN)
IF(TIME .GT. T2) CONST=1./TDELAY*(TOUT-TIME)
900 FCNTRL=FCMAX*CONST
FCYNR=-FCNTRL*SGN*DSIN(PHIPD)
FCZNR=FCNTRL*SGN*DCOS(PHIPD)
CNR ON = .TRUE.
DLTZ=DELTZ
DLTY=0.
C
C COMPUTE CONTROL FORCES AND MOMENTS IN ROLLING AXES
C
1000 CP=DCOS(PHI)
SP=DSIN(PHI)
MCYNR=LT*FCZNR
MCZNR=-LT*FCYNR
FC(2)=FCYNR*CP+FCZNR*SP
FC(3)=-FCYNR*SP+FCZNR*CP
MC(2)=LT*FC(3)
MC(3)=-LT*FC(2)
RETURN
END

```

```

SUBROUTINE TRACKR(TIME,X,XDOT,TNODE)
IMPLICIT REAL *8(A-H,M,O-Z)
INTEGER XODE
DIMENSION X(13),XDOT(13),V(24),VV(3),VV(4),XTG(3),
1VTG(3),LVG(3,3),LVG(3,3),XVEC(3),XNAT(3,3),LVB(3,3),
2LVB(3,3),XVEC2(3),XVEC3(3)
REAL*8 LGV,LVG,LANDAG,LBV,LVB
COMMON/CORF0/HACH,ALPHA,BETA,XL,SAREA,RHO,DIA,TEMP,XCG
1,KCP,SLAT,GAMMA,KAIR,RTUR,HANT,TBEAM
COMMON/COR60/TO,TP,HIN
COMMON/COR70/DELTA,LANLAG,DELGDT,GLAMD,TXV(3),VTX(3)
COMMON/COR170/ZG,YG,ZSDOT,YGDOT
COMMON/COR180/XG(3),YG(3),DEGDEL,DEGLAM,XNR,XNR
COMMON/COR200/DD,DHX
COMMON/COR220/R,RDOT,ZZG,YYG,ZGDT,YGDT,D,DHAX
IF(TIME-LE. TO)GO TO 200
DO 100 I=1,3
100 TXV(I)=TXV(I)+VTX(I)*(TIME-TLAST)
GO TO 300
200 TLAST=TO
300 CONTINUE
C
C
C COMPUTE THE EULER ANGLES OF THE GONIOMETER SYSTEM
RHOG=DSQRT(XTV(1)*XTV(1)+XTV(2)*XTV(2)+XTV(3)*XTV(3))
DELTA=DASIN(-XTV(3)/RHOG)
LANDAG=ATAN2(XTV(2),XTV(1))
DEGDEL=DELTA*57.2957795100
DEGLAM=LANLAG*57.2957795100
CONST=1.D0/DSQRT(1.D0-(XTV(3)/RHOG)**2)
RHDGT=(XTV(1)*VTX(1)+XTV(2)*VTX(2)+XTV(3)*VTX(3))/RHOG
DELGDT=CONST*(-VTX(3)/RHOG+XTV(3)*RHDGT/RHOG/RHOG)
CONST=1.D0/(1.D0+(XTV(2)/XTV(1))**2)
GLAMD=CONST*(VTX(2)/XTV(1)-XTV(2)*VTX(1)/XTV(1)/XTV(1))
IF(TNODE-LT. 1)RETURN
C
C TRANSFORM THE POSITION AND VELOCITY OF THE MISSILE
FROM THE INERTIAL TO THE GONIOMETER SYSTEM
XVEC(1)=0.
XVEC(2)=DELGDT
XVEC(3)=GLAMD
XREF=-RTUR*DCOS(LANDAG)
YREF=-RTUR*SIN(LANDAG)
ZREF=-HANT
VREFX=RTUR*COS(LANDAG)*GLAMD
VREFY=-RTUR*DCOS(LANDAG)*GLAMD
VREFZ=0.
XV(1)=X(1)-XREF
XV(2)=X(2)-YREF

```



```

SUBROUTINE INTERP(A,B,C,D,N,NSLIP)
IMPLICIT REAL*4 (A-H,O-Z)
DIMENSION C(N),D(N)
NSTOP=1
IF(A-LE.C(1).OR.A-GE.C(N))GO TO 300
DO 100 I=1,N
IF(A-LE.C(I).AND.A-LE.C(I+1)) GO TO 200
100 CONTINUE
200 X=C(I)
Y=D(I)
X3=C(I+1)
Y3=D(I+1)
X2=X1-X
Y2=Y1-Y
X3=X1-A
Y3=Y1-Y2
X4=Y2*X3/X2
Y4=Y1-X4
B=Y3
DO 300 J=1,600
300 IF(A-LE.C(I))GO TO 500
PRINT 400,A,N,C(N)
400 FORMAT(1H1,3X,.....,/,5X,'THE NUMBER LIES OUTSIDE THE'
1,.....,/,5X,'A=1,G12.5,5X,C('
2,'RANGE OF TABULATED DATA',/,5X,'.....,
312,')=1,G12.5,/,5X,.....,
4,.....)
NSTOP=-1
GO TO 600
500 PRINT 400,A,NSTOP,C(1)
600 NSTOP=-1
CONTINUE
RETURN
END

SUBROUTINE FILDE(X,XT)
REAL*8 X,XT
DIMENSION X(3),XT(3,3)
DO 5 I=1,3
DO 5 J=1,3
XT(I,J)=0.0
XT(I,2)=-X(I)
XT(1,3)=X(2)
XT(2,1)=X(1)
XT(2,3)=-X(1)
XT(3,1)=-X(2)
XT(3,2)=X(1)
RETURN
END

SUBROUTINE HMAN(V,B,BI,COSAL1,SINAL1,COSAL2,SINAL2,
COSAL3,SINAL3)
IMPLICIT REAL*4 (A-H,O-Z)
DIMENSION R(3,3),BT(3,3),V(24)
DIMENSION B1(3,3),B2(3,3),B3(3,3),T41(3,3)
DO 106 II=1,3
DO 106 JJ=1,3
B1(II,JJ)=0.0D0
B2(II,JJ)=0.0D0
B3(II,JJ)=0.0D0
COSAL1=DCOS(V(1))
SINAL1=DSIN(V(1))
COSAL2=DCOS(V(2))
SINAL2=DSIN(V(2))
COSAL3=DCOS(V(3))
SINAL3=DSIN(V(3))
B1(1,1)=1.0D0
B1(2,2)=COSAL1
B1(3,3)=COSAL1
B1(2,3)=SINAL1
B1(3,2)=-SINAL1
B2(2,2)=1.0D0
B2(1,1)=COSAL2
B2(3,3)=COSAL2
B2(1,3)=-SINAL2
B2(3,1)=SINAL2
B3(1,1)=COSAL3
B3(1,2)=SINAL3
B3(2,1)=-SINAL3
B3(2,2)=COSAL3
B3(3,3)=1.0D0
CALL MATMPY(B3,B2,T41)
CALL MATMPY(B1,T41,B)
CALL TRANSP(B,BT,3,3)
RETURN
END

SUBROUTINE MSUB(A,B,D)
DOUBLE PRECISION A,B,D
DIMENSION A(3,3),B(3,3),
DO 5 I=1,3
DO 5 J=1,3
D(I,J)=0.0D0
5 D(I,J)=A(I,J)-B(I,J)
RETURN
END

```

```

SUBROUTINE MATINV(A,B,DET)
  DIMENSION A(3,3),B(3,3)
  DOUBLE PRECISION A,B,DET,D,DABS,ABSD
  DO 5 I = 1,3
  DO 5 J = 1,3
  5 B(I,J) = 0.0D0
  D=A(1,1)*A(2,2)*A(3,3)-A(3,3)*A(2,1)*A(1,2)+A(1,2)*A(3,1)*A(2,3)-A(2,3)*A(3,1)*A(1,2)+A(2,1)*A(3,2)-A(3,2)*A(2,2)*A(1,3)
  ABSD=DABS(D)
  IF(ABSD.LT. 1.E-05)GO TO 100
  B(1,1) = (A(2,2)*A(3,3)-A(3,3)*A(2,1))/D
  B(1,2) = (A(3,2)*A(1,3)-A(1,2)*A(3,3))/D
  B(1,3) = (A(1,2)*A(2,3)-A(2,2)*A(1,3))/D
  B(2,1) = (A(3,1)*A(2,3)-A(2,1)*A(3,3))/D
  B(2,2) = (A(1,1)*A(3,3)-A(3,1)*A(1,3))/D
  B(2,3) = (A(2,1)*A(1,3)-A(1,1)*A(2,3))/D
  B(3,1) = (A(2,1)*A(3,2)-A(3,1)*A(2,2))/D
  B(3,2) = (A(3,1)*A(1,2)-A(1,1)*A(3,2))/D
  B(3,3) = (A(1,1)*A(2,2)-A(2,1)*A(1,2))/D
  DET = D
  GO TO 999
100 WRITE(6,1000)
1000 FORMAT(1H1,5X,'THE MATRIX IS SINGULAR')
999 CONTINUE
  RETURN
  END

SUBROUTINE THRUST(TIME,FORCE,T,F,N)
  DOUBLE PRECISION TIME,FORCE,T,F
  DIMENSION F(50),T(50)
  IF(TIME.LT. T(1).OR. TIME.GT. T(N)) GO TO 300
  DO 100 I=1,N
  IF(TIME.GE. T(I).AND. TIME.LE. T(I+1))GO TO 200
100 CONTINUE
  200 FORCE=F(I+1)-((T(I+1)-TIME)/(T(I+1)-T(I)))*(F(I+1)-F(I))
  GO TO 400
  300 FORCE=0.0D0
  400 CONTINUE
  RETURN
  END

SUBROUTINE CMAT(V,C,CT,SPSI,CPSI,STHETA,CTHETA,
  -SPHI,CPHI)
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION C(3,3),CT(3,3),V(24),TM1(3,3),TM2(3,3),
  -TM3(3,3),TM4(3,3)
  DO 100 II = 1,3
  DO 100 JJ = 1,3
  TM2(II,JJ) = 0.0D0
  TM1(II,JJ) = 0.0D0
  100 TM3(II,JJ) = 0.0D0
  SPSI = DSIN(V(6))
  CPSI = DCOS(V(6))
  STHETA=DSIN(V(5))
  CTHETA = DCOS(V(5))
  SPHI = DSIN(V(4))
  CPHI = DCOS(V(4))
  TM1(1,1) = 1.0D0
  TM1(2,2) = CPHI
  TM1(2,3) = SPHI
  TM1(3,2) = -SPHI
  TM1(3,3) = CPHI
  TM2(1,1) = CTHETA
  TM2(1,3) = -STHETA
  TM2(2,2) = 1.0D0
  TM2(3,3) = CTHETA
  TM2(3,1) = STHETA
  TM3(1,1) = CPSI
  TM3(1,2) = -SPSI
  TM3(2,1) = SPSI
  TM3(2,2) = CPSI
  TM3(3,3) = 1.0D0
  CALL MATMPY(TM2,TM3,TM4)
  CALL MATMPY(TM1,TM4,C)
  CALL TRANSP(C,CT,3,3)
  RETURN
  END

SUBROUTINE TRANSP(A,B,NRA,NCA)
  DOUBLE PRECISION A,B
  DIMENSION A(NRA,NCA),B(NCA,NRA)
  DO 5 J = 1,NCA
  DO 5 I = 1,NCA
  B(J,I) = 0.0D0
  5 B(J,I) = A(I,J)
  RETURN
  END

```

```

SUBROUTINE CAVEC(X,Z,CONST)
DOUBLE PRECISION X, Z,CONST
DIMENSION X(3), Z(3)
DO 5 I = 1,3
  Z(I) = 0.000
  Z(I) = CONST*X(I)
RETURN
END

```

```

SUBROUTINE MATPY(A,B,C)
REAL*4 A,B,C
DIMENSION A(3,3),B(3,3),C(3,3)
DO 10 I = 1,3
  DO 10 J = 1,3
    C(I,J) = 0.0
    DO 10 K = 1,3
      C(I,J) = C(I,J) + A(I,K)*B(K,J)
    RETURN
  END

```

```

SUBROUTINE VSUB(X,Y,DVEC)
DOUBLE PRECISION X,Y, DVEC
DIMENSION X(3),Y(3), DVEC(3)
DO 5 I = 1,3
  DVEC(I) = 0.000
  DVEC(I) = X(I) - Y(I)
RETURN
END

```

```

SUBROUTINE MAD(A,B,D)
DOUBLE PRECISION A,B, D
DIMENSION A(3,3),B(3,3), D(3,3)
DO 5 I = 1,3
  DO 5 J = 1,3
    D(I,J) = 0.000
    D(I,J) = A(I,J) + B(I,J)
  RETURN
END

```

```

SUBROUTINE VAD(X,Y,DVEC)
DOUBLE PRECISION X,Y, DVEC
DIMENSION X(3),Y(3), DVEC(3)
DO 5 I = 1,3
  DVEC(I) = 0.000
  DVEC(I) = X(I) + Y(I)
RETURN
END

```

```

SUBROUTINE MATVV(A,V,U)
REAL*4 A,V,U
DIMENSION A(3,3),V(3), U(3)
U(1) = A(1,1)*V(1) + A(1,2)*V(2) + A(1,3)*V(3)
U(2) = A(2,1)*V(1) + A(2,2)*V(2) + A(2,3)*V(3)
U(3) = A(3,1)*V(1) + A(3,2)*V(2) + A(3,3)*V(3)
RETURN
END

```

```

SUBROUTINE CMAT(A,C,CONST)
DOUBLE PRECISION A, C,CONST
DIMENSION A(3,3), C(3,3)
DO 5 I = 1,3
  DO 5 J = 1,3
    C(I,J) = 0.000
    C(I,J) = CONST*A(I,J)
  RETURN
END

```

APPENDIX B

LISTING OF SIMPLIFIED MISSILE LAUNCHER

(MISSLNCH) SIMULATION PROGRAM.

On the following pages is a listing of the missile launcher (MISSLNCH) computer program. MISSLNCH is used to simulate the missile dynamics during the launch phase of the analysis in order to obtain initial conditions for the flight phase simulation (MISSIM) program. The computer code developed is compatible with the FORTRAN G, FORTRAN H, and WATFIV compilers used in conjunction with the IBM 370/158 digital computer available at Auburn University. All computations in the program use double precision arithmetic.

All input into the program is accomplished by punched cards. Each card contains ten data fields which are eight columns wide. A generalized (G) field descriptor is used in the format for all read statements. This format was chosen for simplicity, since it allows integer, logical, or real (external fixed point or floating point) variables to be read with the same format. The entire input necessary to use MISSLNCH is read in the main program and is explained below.

IS	Spin-axis moment of inertia (kg-m^2).
IT	Transverse-axis moment of inertia (kg-m^2).
THETAT	Elevation (or pitch) angle of launch tube (rad).
OMEGAT	Steady angular velocity of launch turret measured about vertical axis (rad/sec).
XA,YA	X and Y components, respectively (measured in the turret reference system), of any point fixed within the launch tube relative to a point on the turret axis of rotation (m).

DIA	Missile largest diameter (m).
MASSR	Missile mass (kg).
GRAV	Magnitude of gravity (m/sec ²).
FT	Steady thrust magnitude which acts on missile (Nt).
TX	Spin torque applied to missile during launch (Nt-m).
XETO	The distance from any point fixed within the launch tube to the missile center of mass when tipoff has ended (m).
TODIST	The distance that the missile travels during the tipoff interval (m).
XGT,ZGT	The X and Z components, respectively (measured in the tube reference system), of point G, the center of the infrared angle sensor (m).
X	The vector of initial conditions of the state variables.
PRMT(1)	Time at the beginning of the simulation (sec).
PRMT(2)	Time at the end of the simulation (sec).
PRMT(3)	The maximum allowed integration step size (sec).
PRMT(4)	An upper error round to test for convergence in the integration algorithm.
ERROR	A vector containing error weights of the state variables. Is used to test for convergence in the integration algorithm (The sum of the components must equal 1).

Since this computer program is to be used only during the short time during the launch phase, the missile physical model is considerably more simple than that used in the flight phase. During the launch phase, the missile is modeled as a rigid, constant mass body with constant moments of inertia, which is acted upon by a constant thrust and initial spin torque. In addition, the MISSLNCH program which follows is also a simple mathematical model of the missile/launcher system, and does not contain the level of

generality, nor the inclusion of large numbers of comment cards to explain the sections of the program that was evident in the MISSIM program of Appendix A. If a more complex missile/launcher model is desired, the reader is directed to Ref. 2 for a listing of the more complicated mathematical model used during that study.

Some subroutines used in MISSLNCH are also used in MISSIM. In order to avoid repetition, their listings are presented only in Appendix A. The subroutines are TILDE, TRANSP, MATINV, MATMPY, MATXV, CXMAT, CXVEC, VAC, VSUB, MAD and MSUB.

```

      READ 100, (PRMT(I), I=1,4), (ERROR(I), I=1,13)
      CALL DRKGS (PRMT,X,EZ,GR,13,IHLF,FUNC,OUTPUT,AUX)
      IF (IHLF .LE. 10) STOP
      PRINT 300,IHLF
300  FORMAT(1H ,///,5X,'IHLF=',I2)
      STOP
      END

```

```

      SUBROUTINE VECTOR(A,B,M)
      REAL*8 A(13),B(3)
      DO 100 I=1,3
100  B(I)=A(I+M)
      RETURN
      END

```

```

      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL FUNC, OUTPUT
      REAL*8 IR,LS,IS,IT,LE,MASSR
      COMMON/CON1/IR(3,3),ASTT(3,3),OMEGTT(3),EZ(3,3),E1
1(3,3),ANT(3),DT(3),TSR(3),PTR(3),FET(3),ART(3,3),C
2(3,3),L(3,3),ZGT(3),LE,XBTO,XETO,MASSR,GRAV,IX,FT
      DIMENSION X(13),ERROR(13),PRMT(5),AUX(8,13)
      READ 100,IS,IT,THETAT,OMEGAT,XA,YA,DIA,LE,MASSR,GRAV,
1FT,IX,XETO,ZODIST,XGT,ZGT
100  FORMAT (10G8.0)
      READ 100, (X(I), I=1,13)
      XBTO=1.001+LE-TODIST
      DO 200 I=1,3
      RAT(I)=0.
      OMEGTT(I)=0.
      FET(I)=0.
      PTR(I)=0.
      TSR(I)=0.
      DT(I)=0.
      EGT(I)=0.
      DO 200 J=1,3
      IR(I,J)=0.
      ASTT(I,J)=0.
      EZ(1,J)=J.
      E1(I,J)=0.
      ART(I,J)=0.
      C1(I,J)=0.
200  L(I,J)=0.
      IR(1,1)=LS
      IR(2,2)=IT
      IR(3,3)=IT
      CTHETT=DCOS(THETAT)
      SFEITT=DSIN(THETAT)
      ASTT(1,1)=CTHETT
      ASTT(2,2)=1.
      ASTT(1,3)=-STHETT
      ASTT(3,3)=CTHETT
      ASTT(3,1)=STHETT
      OMEGTI(3)=OMEGAT
      EZ(2,2)=1.
      EZ(3,3)=1.
      RAT(1)=XA
      RAT(2)=YA
      E1(1,1)=1.
      DIA(3)=5*DIA
      TSR(1)=TX
      PTR(1)=PT
      RAT(1)=XGT
      RAT(3)=ZGT

```

```

SUBROUTINE FUNC (TIME,X,XDOT)
  IMPLICIT REAL*8 (A-H,M,O-Z)
  REAL*8 L,LINV,IR,MASSR,LE,LE,LS
  DIMENSION X(13),XDOT(13),LINV(3,3),ATR(3,3),VEC1(3),
  VEC2(3),VEC3(3),MAT1(3,3),MAT2(3,3),MAT3(3,3),ATST(3,
  2,3),EULER(3),RG(3),ANGRAT(3),VEL(3)
  DIMENSION R(3)
  COMMON/CON1/IR(3,3),ASTI(3,3),ONEGT(3),E23(3,3),E1
  1(3,3),RAT(3,3),DT(3),TSR(3),FTR(3),ART(3,3),C
  2(3,3),L(3,3),RGT(3),LE,XBTO,XETO,MASSR,GRAV,TA,FT
  LS=IR(1,1)
  IT=IR(2,2)
  STHET=ASTI(3,1)
  CTHTT=ASTI(1,1)
  SPHI=DSIN(X(7))
  CPHI=DCOS(X(7))
  C(1,1)=1.
  C(1,2)=X(4)*SPHI
  C(1,3)=X(4)*CPHI
  C(2,2)=CPHI
  C(2,3)=-SPHI
  C(3,2)=SPHI
  C(3,3)=CPHI
  ART(1,1)=1.
  ART(1,2)=X(9)
  ART(1,3)=-X(4)
  ART(2,1)=-CPHI*X(9)+SPHI*X(8)
  ART(2,2)=CPHI
  ART(2,3)=SPHI
  ART(3,1)=SPHI*X(9)+CPHI*X(8)
  ART(3,2)=-SPHI
  ART(3,3)=CPHI
  CALL MATXV(ART,FTR,VEC2)
  RAT(1)=-MASSR*GRAV*STHET+VEC2(1)
  RAT(2)=VEC2(2)
  RAT(3)=MASSR*GRAV*CTHET+VEC2(3)
  CALL TRANSP(ATST,ATST,3,3)
  CALL TRANSP(ATR,ATR,3,3)
  XDOT(4)=X(13)
  XDOT(5)=0.
  XDOT(6)=0.
  CALL VECTOR(X,VEC1,9)
  CALL TILDE(VEC1,MAT1)
  CALL VECTOR(X,VEC1,3)
  CALL MATXV(MAT1,VEC1,VEC2)
  CALL VECTOR(XDOT,VEC1,3)
  CALL VAD(VEC1,VEC2,VEC3)
  CALL MATXV(ATR,VEC3,VEC1)
  DO 100 I=1,3
    XDOT(I)=VEC1(I)
  100 XDOT(I)=VEC1(I)

  SUBROUTINE FUNC (TIME,X,XDOT)
  IF (X(1).LE. XBTJ) GO TO 200
  XDOT(7)=X(10)
  XDOT(10)=TX/IS
  JNEGAT=ONEGT(1)
  XDOT(14)=PT/MASSR-GRV*STHET+XT*(ONEGT*CTHET)**2
  XDOT(8)=0.
  XDOT(9)=0.
  XDOT(11)=0.
  XDOT(12)=0.
  RETURN
  200 IF (X(1).LT. XETJ) GO TO 800
  CALL VSUB(RAT,RGT,VEC1)
  CALL MATXV(ASST,VEC1,VEC2)
  CALL VECTOR(X,VEC1,0)
  CALL VAD(VEC1,VEC2,VEC3)
  DO 300 I=1,3
    R(I)=VEC3(I)
  300 R(I)=VEC3(I)
  RAD=57.2957795100
  PSIT=ONEGT(3)*TIME
  CP=CCOS(PSIT)
  SP=DSIN(PSIT)
  MAT1(1,1)=CP
  MAT1(2,2)=CP
  MAT1(1,2)=SP
  MAT1(2,1)=-SP
  MAT1(3,1)=0.
  MAT1(1,3)=0.
  MAT1(2,3)=0.
  MAT1(3,2)=0.
  MAT1(3,3)=1.
  CALL MATMPY(ASST,MAT1,MAT2)
  CALL TRANSP(MAT2,MAT3,3,3)
  CALL MATXV(MAT3,RG,VEC1)
  DO 400 I=1,3
    R(I)=VEC1(I)
  400 R(I)=VEC1(I)
  R(1)=R(1)-.95*CP
  R(2)=R(2)-.95*SP
  R(3)=R(3)-4.5
  CALL MATMPY(MAT,MAT2,MAT1)
  EULER(1)=DATAN2(MAT1(2,3),MAT1(3,3))*RAD
  EULER(2)=DARSIN(-MAT1(1,3))*RAD
  EULER(3)=DATAN2(MAT1(1,2),MAT1(1,1))*RAD
  CALL MATMPY(ART,ASTI,MAT2)
  CALL MATXV(MAT2,ONEGT,VEC1)
  DO 500 I=1,3
    ANGRAT(I)=X(14)+VEC1(I)
  500 ANGRAT(I)=X(14)+VEC1(I)
  CALL VECTOR(XDOT,VEC1,0)
  CALL MATXV(ART,VEC1,VEC3)
  CALL TILDE(ONEGT,MAT1)

```



```

CALL MATVV(MAT1,RAT,VEC2)
CALL MATVV(ASST,VEC2,VEC1)
CALL MATIV(ART,VEC1,VEC2)
CALL VAD(VEC2,VEC3,VEC1)
DO 600 I=1,3
600 VEL(4)=VEC1(I)
PRINT 700, TIME, RG(1), R(1), EULER(1), ANGRAT(1), VEL(1), R
1G(2), R(2), EULER(2), ANGRAT(2), VEL(2), RG(3), R(3), EULER(
23), ANGRAT(3), VEL(3)
700 FORNAT('H //, 1X, 'LINE=', G12.5, //, 1H, 3X, 'RG=', G12.5,
13X, 'X=', G12.5, 3X, 'PHI =', G12.5, 3X, 'P=', G12.5, 3X, 'U=',
2G12.5, //, 1H, 3X, 'YG=', G12.5, 3X, 'Y=', G12.5, 3X, 'THETA=',
3G12.5, 3X, 'Q=', G12.5, 3X, 'V=', G12.5, 3X, //, 1H, 3X, 'ZG=',
4G12.5, 3X, 'Z=', G12.5, 3X, 'PSI=', G12.5, 3X, 'R=', G12.5, 3X,
5'N=', G12.5)
STOP
800 CONTINUE
CALL VECTOR(X,VEC1,9)
CALL MATIV(C,VEC1,VEC2)
DO 900 I=1,3
900 XDOT(I+6)=VEC2(I)
CALL VECTOR(X,VEC1,0)
CALL MATIV(ATST,VEC1,VEC2)
CALL VAD(RAT,VEC2,VEC1)
CALL TILDE(OMEGST,MAT1)
CALL MATIV(MAT1,VEC1,VEC2)
CALL MATIV(MAT1,VEC2,VEC1)
CALL VECTOR(XDOT,VEC2,0)
CALL MATIV(ATST,VEC2,VEC3)
CALL MATIV(MAT1,VEC3,VEC2)
CALL CXVEC(VEC2,VEC3,2,D0)
CALL VAD(VEC1,VEC3,VEC2)
CONST=1./MASSR
CALL CXVEC(FRT,VEC3,CONST)
CALL VSUB(VEC3,VEC2,VEC1)
CALL MATIV(E1,VEC1,VEC2)
XDOT(13)=VEC2(1)
CALL VECTOR(X,VEC1,0)
CALL MATIV(ATST,VEC1,VEC2)
CALL VAD(VEC2,RAT,VEC1)
CALL MATIV(MAT1,VEC1,VEC2)
CALL MATIV(MAT1,VEC2,VEC1)
CALL MATIV(ASST,VEC1,VEC3)
CALL VECTOR(X,VEC1,3)
CALL VECTOR(X,VEC2,9)
CALL TILDE(VEC2,MAT2)
CALL MATIV(MAT2,VEC1,VEC2)
CALL MATIV(ATR,VEC2,VEC1)
CALL MATIV(MAT1,VEC1,VEC2)
CALL MAD(IR,MAT1,L)

```



```

      DO 2400 I=1,NDIM
      DELT=DELT+AUX(8,I)*DABS(AUX(9,I)-Y(I))
      IF (DELT-PRMT(4)) 2800,2800,2500
C
C      2400 I=1,NDIM
C      2500 IF (IHLF-10) 2600,3600,3600
C      2600 DO 2700 I=1,NDIM
C      2700 AUX(4,I)=AUX(5,I)
C      ISTEP=ISTEP+ISTEP-4
C      Y=X-H
C      IEND=0
C      GOTO 1600
C
C      RESULT VALUES ARE GOOD
C      2800 CALL FCT(X,Y,DERV)
C      DO 2900 I=1,NDIM
C      AUX(1,I)=Y(I)
C      AUX(2,I)=DERV(I)
C      AUX(3,I)=AUX(6,I)
C      Y(I)=AUX(5,I)
C      DERY(I)=AUX(7,I)
C      CALL OUTP(X,H,Y,DERV,IHLF,NDIM,PRMT)
C      IF (PRMT(5)) 4000,3000,4000
C      DO 3100 I=1,NDIM
C      Y(I)=AUX(1,I)
C      DERY(I)=AUX(2,I)
C      IREC=IHLF
C      IF (IEND) 3200,3200,3900
C
C      3200 INCREMENT GETS DOUBLED
C      IHLF=IHLF-1
C      ISTEP=ISTEP/2
C      H=H+H
C      IF (IHLF) 400,3300,3300
C      IMOD=ISTEP/2
C      IF (ISTEP=IMOD-IMOD) 400,3400,400
C      3400 IF (DELT-.0200*PRMT(4)) 3500,3500,400
C      3500 IHLF=IHLF-1
C      ISTEP=ISTEP/2
C      H=H+H
C      GOTO 400
C
C      RETURNS TO CALLING PROGRAM
C      3600 IHLF=1)
C      CALL FCT(X,Y,DERV)
C
C      GOTO 3400
C      3700 IHLF=12
C      GOTO 3900
C      3800 IHLF=13
C      3900 CALL OUTP(X,Y,DERV,IHLF,NDIM,PRMT)
C      4000 RETURN
C      END

```

```

      IF (PRMT(5)) 4000,800,4000
      300 ITEST=0
      900 ISTEP=ISTEP+1
C
C      START OF INNERMOST RUNGE-KUTTA LOOP
C      J=1
C      1000 AJ=A(J)
C      BJ=B(J)
C      CJ=C(J)
C      DO 1100 I=1,NDIM
C      R1=H*DERV(I)
C      R2=AJ*(R1-BJ*AUX(6,I))
C      Y(I)=Y(I)+R2
C      R2=R2+R2*R2
C      1100 AUX(6,I)=AUX(6,I)+R2-CJ*R1
C      IF (J-4) 1200,1500,1500
C      1200 J=J+1
C      IF (J-3) 1300,1400,1300
C      1300 I=X+.5D0*H
C      1400 CALL FCT(X,Y,DERV)
C      GOTO 1000
C      END OF INNERMOST RUNGE-KUTTA LOOP
C
C      TEST OF ACCURACY
C      1500 IF (ITEST) 1600,1600,2000
C
C      IN CASE ITEST=0 THERE IS NO POSSIBILITY FOR TESTING
C      OF ACCURACY.
C      1600 DO 1700 I=1,NDIM
C      1700 AUX(4,I)=Y(I)
C      ITEST=1
C      ISTEP=ISTEP+ISTEP-2
C      IHLF=IHLF+1
C      X=X-H
C      H=.5D0*H
C      DO 1900 I=1,NDIM
C      Y(I)=AUX(1,I)
C      DERY(I)=AUX(2,I)
C      1900 AUX(6,I)=AUX(3,I)
C      GOTO 900
C
C      IN CASE ITEST=1 TESTING OF ACCURACY IS POSSIBLE
C      2000 IMOD=ISTEP/2
C      IF (ISTEP=IMOD-IMOD) 2100,2300,2100
C      CALL FCT(X,Y,DERV)
C      DO 2200 I=1,NDIM
C      AUX(5,I)=Y(I)
C      2200 AUX(7,I)=DERV(I)
C      GOTO 900
C
C      COMPUTATION OF TEST VALUE DELT
C      2300 DELT=0.D0

```

APPENDIX C

AERODYNAMIC COEFFICIENTS

In this appendix, aerodynamic data for a typical SHORADS rocket, the motion of which was simulated in this study, is presented and the way in which this data is used in the flight simulation code is described. This data was provided by Mr. John Howerton, U.S. Army Research and Development Command, Redstone Arsenal, Alabama.

C_N and C_y

The coefficients C_N and C_y are obtained from C_{N_α} for which values are given in Table C-1a for twenty-seven Mach number values and two incidence angle ($i_T = \sqrt{\alpha^2 + \beta^2}$) ranges, $i_T \leq 4^\circ$ and $i_T \geq 4^\circ$. During the simulation, when values of C_N and C_y are needed, the incidence angle is computed and the proper one-dimensional array of stored C_{N_α} values is chosen. Corresponding Mach numbers which bracket the flight Mach number are determined next and the value of C_{N_α} used in computing C_N and C_y is determined by linear interpolation. For the rocket considered herein, C_N and C_y for $i_T > 4^\circ$ are computed using $C_N = C_{N_0} + C_{N_\alpha} \alpha$ and $C_y = -C_{N_0} - C_{N_\alpha} \beta$, where C_{N_0} is a function of Mach number as shown by Table C-1a.

C_{A_0}

Values of the axial force coefficient, C_{A_0} , as a function of Mach number are given in Table C-1a also. A linear interpolation subroutine contained in the simulation code is used to determine the proper value of C_{A_0} .

TABLE C-1. AERODYNAMIC CHARACTERISTICS.

M	C_{N_α}	C_{N_α}	C_{N_o}	C_{A_o}
	$i_t < 4^\circ$	$i_t > 4^\circ$	$i_t > 4^\circ$	
0.0	15.0	26.0	-0.8	0.44
0.1	15.0	26.0	-0.8	0.44
0.2	15.0	26.0	-0.8	0.433
0.3	15.6	26.1	-0.79	0.430
0.4	15.0	26.1	-0.78	0.42
0.5	15.5	26.4	-0.74	0.416
0.6	16.0	26.5	-0.70	0.41
0.7	16.25	26.7	-0.71	0.41
0.8	16.2	26.5	-0.72	0.41
0.85	16.25	26.7	-0.725	0.415
0.90	16.7	27.0	-0.74	0.435
0.925	17.1	27.5	-0.74	0.46
0.95	17.5	27.6	-0.74	0.50
0.975	17.7	27.8	-0.725	0.55
1.0	18.0	28.0	-0.715	0.605
1.025	18.2	28.3	-0.71	0.645
1.05	18.5	28.5	-0.705	0.665
1.075	18.6	28.6	-0.70	0.685
1.1	19.0	28.7	-0.70	0.69
1.15	19.1	28.9	-0.70	0.68
1.2	18.9	28.5	-0.70	0.66
1.25	18.8	28.0	-0.70	0.605
1.3	18.6	27.5	-0.695	0.57
1.4	18.5	27.0	-0.695	0.55
1.5	18.0	26.2	-0.625	0.54
1.6	18.0	25.5	-0.550	0.53
1.7	18.0	25.0	-0.480	0.53

(a)

TABLE C-1 (CONT)

108

i_T (DEGREES)	C_{fp} $M \leq 1$	C_{fp} $M > 1$
0.0	-10.81	-9.406
1.0	-11.80	-10.56
2.0	-12.51	-11.59
3.0	-12.99	-12.52
4.0	-13.30	-13.32
5.0	-13.50	-14.02
8.0	-13.96	-15.42
10.0	-14.77	-15.78
12.0	-16.54	-15.68

(b)

$l-x_{CC}$ (METERS)	C_{mq} CANARDS DEPLOYED	C_{mq} NO CANARD DEPLOYMENT
0.985	-320.0	-250.0
1.000	-350.0	-300.0
1.110	-370.0	-330.0

(c)

l_r PERCENT ACTUATOR PISTON TRAVEL	B_r PERCENT CANARD DEPLOYMENT
0.0	57.0
5.0	57.0
7.5	58.0
10.0	59.0
15.0	64.25
20.0	69.5
25.0	74.25
30.0	79.0
35.0	82.5
40.0	86.0
45.0	88.25
50.0	90.5
55.0	92.0
60.0	93.5
65.0	95.0
70.0	96.5
75.0	97.25
80.0	98.0
85.0	98.5
90.0	99.0
95.0	99.5
100.0	100.0

(d)

B_r \ M	0.9	1.0	1.05	1.15	1.3	1.5
57	1000.0	968.0	955.0	945.0	942.0	947.0
60	1003.0	974.0	963.0	953.0	951.0	955.0
65	1013.5	998.5	978.5	967.5	964.5	967.0
70	1024.0	1003.0	994.0	982.0	978.0	979.0
75	1039.5	1020.0	1011.0	999.0	994.0	995.0
80	1055.0	1037.0	1028.0	1016.0	1010.0	1012.0
85	1079.0	1057.0	1047.0	1034.0	1027.0	1028.5
90	1104.0	1079.0	1067.0	1054.0	1048.0	1049.0
95	1132.0	1104.0	1090.0	1076.0	1080.0	1070.0
100	1162.0	1125.0	1110.0	1100.0	1096.0	1100.0

 $f_2(B_r, M)$

(e)

f_2 (MILLIMETERS)	i_T (DEGREES)
0.0	0.0
-2.0	1.0
-8.0	2.0
-16.0	3.0
-24.0	4.0
-30.0	5.0
-36.0	7.0
-38.0	9.0
-38.0	12.0

(f)

$$\underline{C_{\ell p}}$$

The roll damping coefficient, $C_{\ell p}$, is given in Table C-1b as a function of incidence angle, i_T , for Mach numbers less than and greater than unity. In the simulation code linear interpolation is used to find the proper $C_{\ell p}$ value.

$$\underline{C_{m q} \text{ and } C_{n r}}$$

The pitch and yaw rate damping coefficients, $C_{m q}$ and $C_{n r}$, respectively, are equal, since the rocket is geometrically symmetric with respect to the x-axis. For canard (destabilizing fins) undeployed and deployed, values of the coefficient $C_{m q}$ are given in Table C-1c along with corresponding values of $\ell - x_{cg}$, the distance of the center of mass of the rocket from its nose. The distance ℓ_{cg} is computed in the simulation code during each evaluation of the derivatives of the state variables and the linear interpolation subroutine is used (as in other calculation discussed above) to find the proper value of $C_{m q} = C_{n r}$.

$$\underline{C_{m \alpha} \text{ and } C_{n \beta}}$$

For the rocket simulated herein, $C_{n \beta} = -C_{m \alpha}$ and $C_{m \alpha}$ is a function of the center of pressure location x'_{cp} (measured from the aft end of the rocket) which in turn is a function of Mach number, aerodynamic incidence angle and B_r , the percent of the total area of the canards (destabilizing fins) which is exposed to the flow field surrounding the rocket. In the simulation code, $C_{m \alpha} = C_{N \alpha} (x'_{cp} - x_{cg})$ and $x'_{cp} = f_1(B_r, M) + f_2(i_T)$. Here, B_r is dependent upon ℓ_r , the distance the deployment piston has traveled. The distance ℓ_r is a linear function of time which is zero at 0.8 seconds

after booster motor ignition and has its maximum value at 1.7 seconds after booster motor ignition.

To find C_{m_α} , the value of ℓ_r is first determined. Then B_r is determined by linearly interpolating using the values in Table C-1d. Next, $f_1(B_r, M)$ is found by using a double linear interpolation scheme and the data of Table C-1e. The value of i_T and the corresponding value for f_2 are then found from the data listed in Table C-1f. These values of f_1 and f_2 are used to find x'_{cp} and the current value of x_{cg} is used along with the x'_{cp} value and the appropriate value of C_{N_α} to compute C_{M_α} .